HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

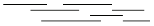
ALGORITHMS FOR GENOME ASSEMBLY

Leena Salmela, Veli Mäkinen, Niko Välimäki, Johannes Ylinen, and Esko Ukkonen

Genome size: 350 Mbp

Melitaea cinxia
Photo: Niclas Fritzén

Reads

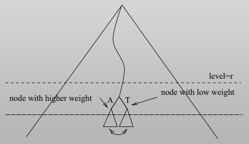


	454	SOLiD	Illumina
Number of reads	10 million	200 million	300 million
Read length	400–800 bp	50 bp	75-150 bp
Errors	Indels	Mismatches	Mismatches
Paired end	-	-	600 bp, 800 bp
Mate pairs	7 kbp, 16 kbp	2 kbp, 3 kbp	1 kbp, 2-4 kbp
Other	-	Color coding	-

Total input data size: 45000 Mbp

Hybrid SHREC

- Based on SHREC by Schröder et al.
- Build a suffix trie of the read set.
- Correct low weight nodes in the trie by comparing to siblings
- Support for simultaneous correction of color coded and base coded reads



L. Salmela: Correction of sequencing errors in a mixed set of reads. *Bioinformatics* 26:10(1284–1290), 2010. (Award for best paper submitted to HiTSeq 2010).

Error Correction

Remove sequencing errors by aligning the reads with each other

Coral

- Build multiple alignments of reads that share *k*-mers
- Correct reads based on these multiple alignments
- Sequencing error model can be specified by setting gap penalty and mismatch penalty for multiple alignments

```
GTAA–GTTGAACCCTTA
  AAAGTTGAACCCTTACC
     GTTGAACC–TTACCCCGG
       GACCCCTTACCCGGTTCA
```

L. Salmela and J. Schröder: Correcting errors in short reads by multiple alignments. *Bioinformatics* 27(11):1455–1461, 2011. (Also in HiTSeq 2011).

MIP Scaffolder

- Cleaning input:
  - Keeping only more reliable mate pairs
  - Bundling mate pairs that connect the same contigs together
  - Estimating the distance between contigs based on the mate pairs
- Partitioning the problem into smaller subproblems of *restricted* size
- Solving each subproblem as a mixed integer program (MIP)

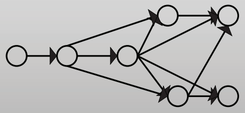
L. Salmela, V. Mäkinen, N. Välimäki, J. Ylinen, and E. Ukkonen: Fast scaffolding with small independent mixed integer programs. *Bioinformatics* 27:23(3259–3265), 2011.

Overlap Computation

Find suffix-prefix overlaps between reads. Represent the overlaps in an overlap graph.

Overlap Tool

- Supports mismatches and indels in the overlaps
- Based on *Burrows-Wheeler transform*, *backward backtracking* (Lam et al. 2008) and *suffix filters* (Kärkkäinen et al. 2008)
- Easy to parallelize
- Scales up to millions of reads



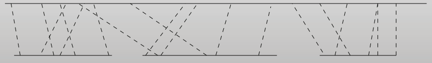
N. Välimäki, S. Ladra, and V. Mäkinen: Approximate all-pairs suffix/prefix overlaps. *Information & Computation* 10.1016/j.ic.2012.02.002. Available online, 2012.

Contig Assembly

Report paths in the overlap graph as contigs, i.e. contiguous sequences.

Validation with ESTs

- Align ESTs against scaffolds:
  - Find local maximal approximate matches (swift by Rasmussen et al. 2006)



  - Produce maximal colinear chains of the above matches (Abouelhoda 2007)
- Compute the coverage of ESTs

Scaffolding

Mate pairs and proteins give links between contigs. Remove minimum number of mate pairs so that the remaining ones are consistent.

(Error corrected)
Mate pairs

Protein links

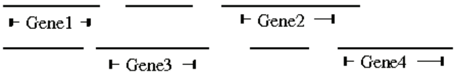
Gap Closing

Use paired end reads to fill the gaps between contigs.

Validation

Genetic map, Map ESTs to scaffolds,...

Annotation



Acknowledgements

Jan Schröder, Simon J. Puglisi, Ilkka Hanski, Rainer Lehtonen, Virpi Ahola, Lars Paulin, Petri Auvinen, Panu Somervuo, Liisa Holm, Patrik Koskinen, Jussi Nokso-Koivisto, Pasi Rastas

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

IDENTIFYING REGULATORY MODULES IN GENOME

Jarkko Toivonen, Department of Computer Science

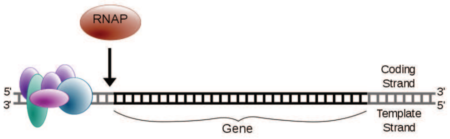
REGULATION OF GENES

The basic question is why gene expression differs between cells of single organism even though the cells contain the same DNA.
What affects the gene expression of a cell?

- Condition
  - For example, increased temperature or stress to cell causes Heat Shock Factor (HSF) to be activated.
- Cell type: neuron, germ, blood cells, etc
- The stage of development of an organism: Embryo, Fetus, Adult, etc

What mechanism regulates the expression of genes?

- A *promoter* is an area in DNA close to the beginning of a gene. Transcription of a gene starts here.
- Certain proteins that chemically bind to this promoter area can regulate the transcription of the gene
- These proteins that bind to DNA and regulate the transcription are called *transcription factors* (TF). They can be either *Activators* or *Repressors*.

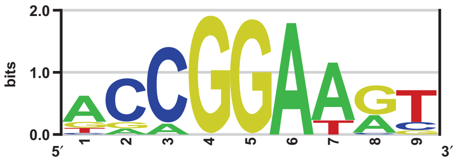


A MODEL FOR A BINDING SITE

Binding sites of transcription factors

- In order to understand how the regulatory system works, it is important to be able to describe and predict the binding sites of transcription factors in the genome
- A model that describes the binding sites where the TF prefers to bind is called *motif*.
- There are several ways to represent a motif:
  - A *consensus sequence* of a TF is the DNA sequence with the highest binding affinity to the TF
  - Regular expression (like ACG[GC]TT)
  - *Position Weight Matrix* (PWM) and its sequence *logo*

An example of a PWM logo for the ERG factor:



DATA

We need a large set of sequences where we know that a fixed transcription factor has bound. From this dataset we want to learn a motif model for the transcription factor in question.
The SELEX procedure (*Systematic evolution of ligands by exponential enrichment*) is a high-throughput *in vitro* method for selecting those sequences that get bound by a TF.

- Starts with a library of random sequences of constant length: for instance 14 or 20 bp
- The proteins are let to bound to the random sequences
- The unbound sequences are removed
- The selected sequences are cloned by PCM
- The selection process is repeated for the cloned sequences
- The selected sequences can be sequenced after each round of selection

Why use SELEX?

- To make high precision motifs, lots of bound sequences are needed
- Fast and relative inexpensive
- Results from several different experiments can be sequenced in parallel using barcoding

LEARNING A PWM FROM SELEX DATA

Using the SELEX data

- The SELEX procedure results in a set of fixed length sequences that were bound by the transcription factor
- The length of the binding site is usually shorter than the length of the SELEX window
- Therefore, the sequences are fed to a motif finding program
- An alignment for the sequences is produced
- An example of counts from the aligment of the SELEX experiment with the ERG transcription factor

	1	2	3	4	5	6	7	8	9
A	164	22	23	0	0	164	164	98	6
C	10	164	164	0	0	1	1	9	42
G	37	23	0	164	164	0	1	164	21
T	31	3	0	0	1	1	40	2	164

- These counts are then normalized columnwise, resulting in a multinomial distribution in each of the columns. This matrix can be visualised as the previously shown sequence logo.

MODEL FOR REGULATORY AREAS

The simple model isn't enough because of cooperation of transcription factors and the chromatin structure of DNA.
Our plan is to create a model for regulatory areas.

- We try to take a simple model for single isolated motif and combine these to create a more complex system that tries to describe the co-operation of a set of transcription factors
- Distances between transcription factors and their orientation can affect the strength of binding.
- This more complicated model can be used to predict clusters of binding sites in the genome
- The validity of the model can be tested with *in vivo* data, like ChIP-seq

CAUSES OF CANCER

Even though understanding of regulatory system is important in it self, still the main objective is cancer research.

- Oncogenes promote cell growth and reproduction
- Tumor suppressor genes inhibit cell division and survival
- Mutations in the DNA can affect the expression of these genes
- This can result in unrestricted growth, i.e. cancer

[1] Arttu Jolma, Teemu Kivioja, Jarkko Toivonen, *et al*. Multiplexed massively parallel selex for characterization of human transcription factor binding specificities. Genome Res. 20, 861–873 (2010).

This is joint work with Arttu Jolma, Teemu Kivioja, Pasi Rastas, Mikko Sillanpää, Jussi Taipale and Esko Ukkonen.

Mining the UKIDSS GPS: star formation and embedded clusters

Otto Solin[1,2], Esko Ukkonen[1], Lauri Haikala[3], Sami Maisala[2]

[1] University of Helsinki, Department of Computer Science, [2] University of Helsinki, Department of Physics, [3] Finnish Centre for Astronomy with ESO

otto.solin@helsinki.fi

Major part of star formation, be it low- or high-mass stars, takes place in clusters. The clusters are not bound and will eventually disrupt e.g. because of the Galactic differential rotation. The stellar clusters trace therefore the recent Galactic star formation. The younger the clusters are the more compact they are and the more closely they are associated with the interstellar gas and dust clouds they formed in. Detailed study of young clusters still associated with their parent cloud will provide information on the star formation process and the stellar initial mass function (IMF).

At the moment some 2000 Galactic stellar clusters are known. This is only a small fraction of the estimated total population of which a major part is obscured by interstellar dust to us and can not be observed in optical wavelengths. However, the extinction decreases at longer wavelengths and already at 2.2 microns in the NIR the extinction in magnitudes is only 11 percent of that in the *V* band.

The aim of this research is to develop methods to locate previously unknown stellar clusters from the UKIDSS Galactic Plane Survey catalogue data release 7.

The search method takes pre-filtered catalogue data, divided into overlapping bins, and performs a maximum likelihood fitting of a mixture of a Gaussian density and a uniform background. On each bin the fitting is done using the standard Expectation Maximization (EM) algorithm. In addition to the UKIDSS GPS catalogue, stars brighter than 10^{m} in *K* from the 2MASS survey are used, because the brighter stars saturate in UKIDSS and moreover tend to produce false positives around them.

Scrutiny of the data base and the survey images reveals that the UKIDSS pipeline source detection algorithm tends to classify most of the objects within regions of variable surface brightness as non-stellar (parameter `mergedClass=+1`), whereas objects with intensity profiles similar to the UKIDSS WFCAM point spread function are classified as star-like (`mergedClass=-1`). Clustering non-stellar sources directs the search to stellar clusters either embedded in or near molecular/dust clouds. Besides stellar clusters, the search targets also the locations of non-clustered star formation and single embedded stars with associated nebulosities. The surface brightness, either due to outflow activity or reflection, will produce "cluster" detections.

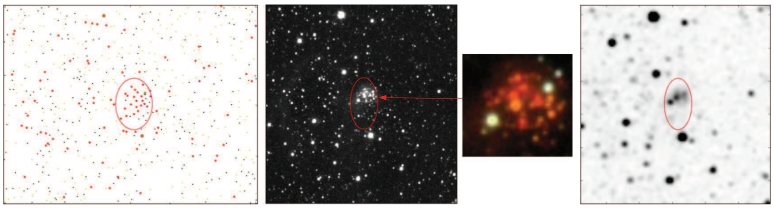
A fraction of the catalogue sources are due to data artefacts. The artefacts cause highly varying extended surface brightness which causes the pipeline to classify most of the sources within the artefact as non-stellar sources. In addition sharp features in the artefacts produce nonexistent sources.

As expected most of the detected new clusters (137) or sites of star formation (30) are tightly concentrated on the Galactic plane. Relatively few new clusters were detected in the direction of the northern Galactic plane.

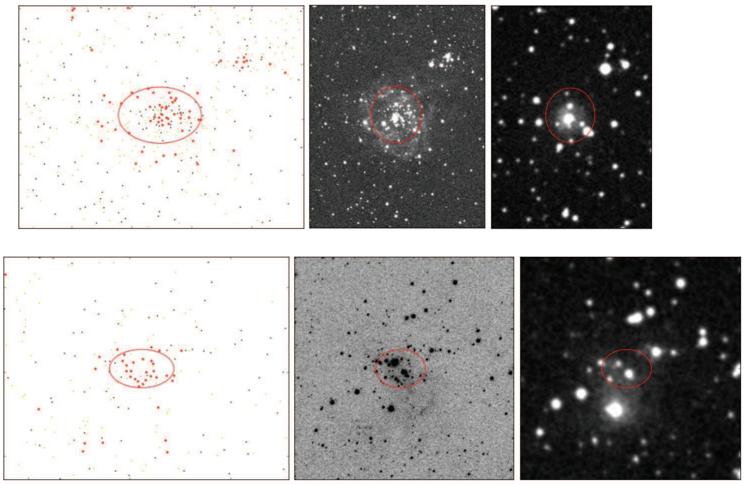
Most images of the new cluster candidate areas show clear signs of reflected light in particular in in the *K* band thus indicating embedded clusters or sites of star formation.

The results are in press for the journal Astronomy & Astrophysics (http://arxiv.org/abs/1203.5292).

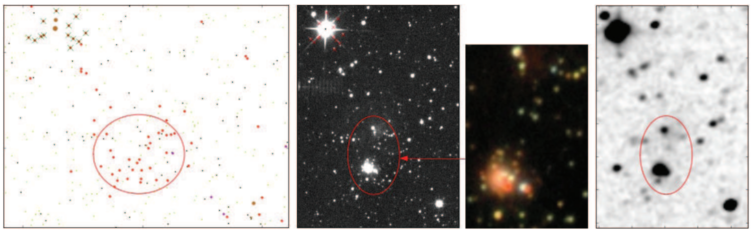
New cluster candidates identified previously as infrared point sources



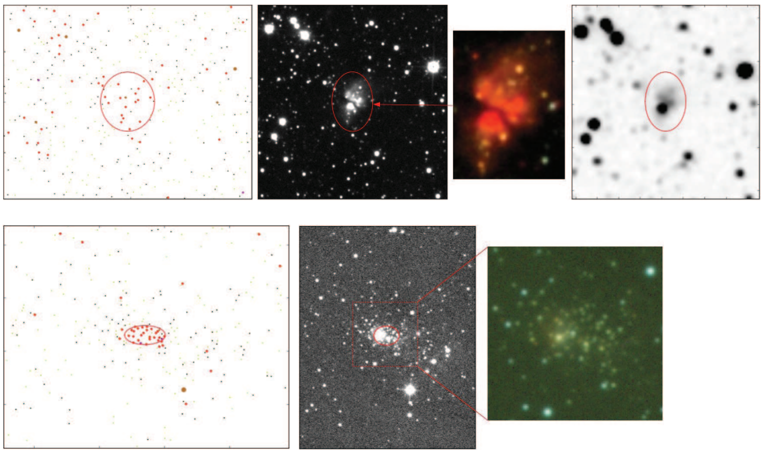
In the leftmost panel are the UKIDSS catalogue entries in the cluster area. The red points are UKIDSS non-stellar sources brighter than 17^{m} in *K*, black points other sources brighter than 17^{m} in *K*, yellow points sources fainter than 17^{m} in *K*, and brown points sources listed in 2MASS but not in UKIDSS GPS. The red confidence ellipse is the cluster area given by the EM-algorithm. In the two middle panels are the *K* band and *JHK* false colour images of the cluster area. In the 2MASS image (the rightmost panel) of the same area no cluster can be seen.



The two candidates above are reflection nebulae in optical images (rightmost panels). The object NW of the candidate in the upper middle panel is either another cluster or part of this larger cluster.



This candidate is not associated with any object in the SIMBAD data base. The bright star in the NE corner of the image causes non-stellar classifications that produce false positive clusters: the algorithm removes the sources overplotted with a cross. In the 2MASS image (the rightmost panel) of the same area no cluster can be seen.



Besides an IRAS point source a millimetre source, a maser and an infrared dark cloud are detected in the direction of the candidate in the upper panels, and towards the candidate in the lower panels an MSX source, an HII region and a submillimetre source.

The number of indicators seen in the direction of many candidates gives confidence the new clusters or embedded star formation locations are real entities and not produced by chance nor are due to catalogue artefacts. In general radio surveys find circumstellar dust envelopes and disks, and cold cores of molecular clouds. In areas where a radio telescope sees only a point source or signs of e.g. an ultracompact HII region, the UKIDSS images show structures of surface brightness and single stars thus verifying the results of the millimetre/submillimetre radio surveys of suspected star forming regions.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

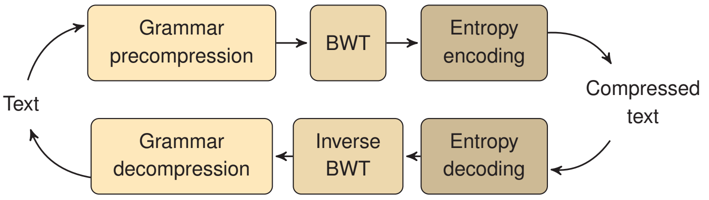
ACCELERATING BURROWS–WHEELER COMPRESSION WITH GRAMMAR PRECOMPRESSION

Juha Kärkkäinen, Pekka Mikkola and Dominik Kempa

The speed of text compressors based on the Burrows–Wheeler transform (BWT) — such as the popular `bzip2` — is limited by the time needed to compute the BWT during compression and its inverse during decompression. We propose to speed up Burrows–Wheeler compression by performing a grammar-based *precompression* before the BWT. We have developed a fast grammar precompressor as a part of an experimental Burrows–Wheeler compressor, and show with experiments that it accelerates compression and decompression without affecting compressibility.

COMPRESSOR OVERVIEW

The grammar precompressor has been implemented as a part of an experimental Burrows–Wheeler compressor [2] that has three main stages:



The idea of the precompressor is to quickly reduce the data before more expensive stages. Grammar compression is well-suited for the task as it can achieve some compression without harming final compressibility. Grammar compression has been studied as a standalone compression method but not as a precompression method before.

We use Yuta Mori's `divsufsort` algorithm [3] for computing the BWT and `mtl-sa-8` algorithm from [1] for computing the inverse.

We use two experimental entropy coders of our own. One compresses well but is relatively slow. The other is fast but does not compress quite as well.

GRAMMAR PRECOMPRESSION

The grammar precompressor performs one or more rounds of the following:

1. Compute the frequencies of symbol pairs by scanning the text.
2. Choose a set of frequent pairs that cannot overlap (see below).
3. Add the rule $X \rightarrow AB$ for each chosen pair AB, where X is a new non-terminal symbol.
4. Replace all occurrences of chosen pairs with the corresponding non-terminal symbols in a single sequential pass over the text.

Pairs A_1B_1 and A_2B_2 can overlap iff $A_1 = B_2$ or $B_1 = A_2$. We avoid pairs that can overlap to ensure that all occurrences of all pairs are replaced.

Occurrences of rare symbols may be replaced by pairs of bytes to free those rare symbols to be used as non-terminals.

Here is an example with two rounds:

Text	Rules added
singing_do_wah_diddy_diddy_dum_diddy_do	$A \rightarrow \texttt{\_d}$, $B \rightarrow \texttt{id}$, $C \rightarrow \texttt{in}$
s*C*g*C*g*A*o_wah*AB*dy*AB*dy*A*um*AB*dy*A*o	$D \rightarrow AB$, $E \rightarrow \texttt{dy}$, $F \rightarrow A\texttt{o}$, $G \rightarrow C\texttt{g}$
s*GGF*_wah*DEDEA*um*DEF*	

The decompressor computes the full expansion of all rules and then replaces all occurrences with a single scan of the text.

Expanded rules
$A \rightarrow \texttt{\_d}$, $B \rightarrow \texttt{id}$, $C \rightarrow \texttt{in}$
$D \rightarrow \texttt{\_did}$, $E \rightarrow \texttt{dy}$, $F \rightarrow \texttt{\_do}$, $G \rightarrow \texttt{ing}$

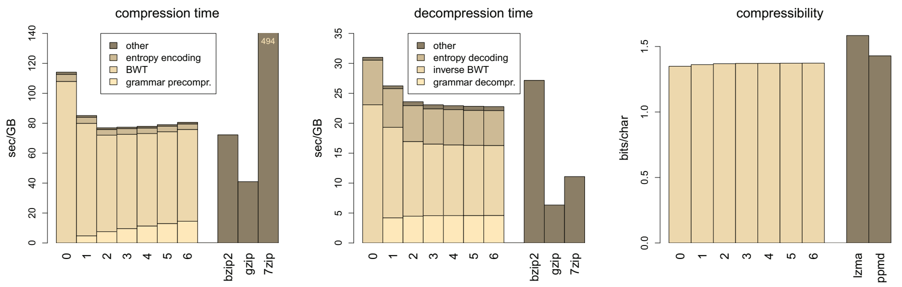
EXPERIMENTS

We ran three sets of experiments to test three hypotheses:

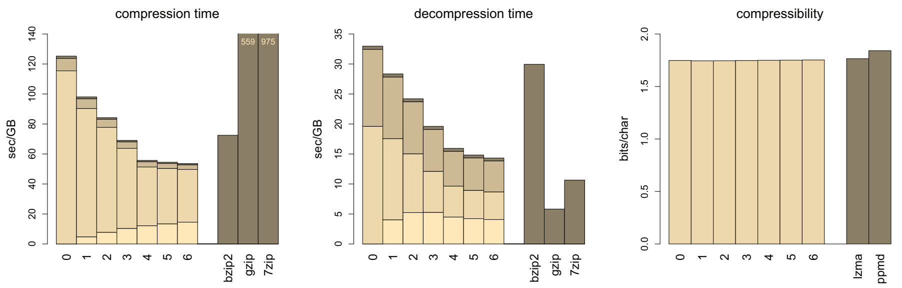
1. Precompression improves the total compression time.
2. Precompression improves the total *de*compression time.
3. Precompression does not hurt the compressibility.

The timing experiments use the fast entropy coder and the compressibility experiments use the slow but good entropy coder. The x-axis labels from 0 to 6 are the number of precompression rounds. Other well-known compressors are included as a reference point.

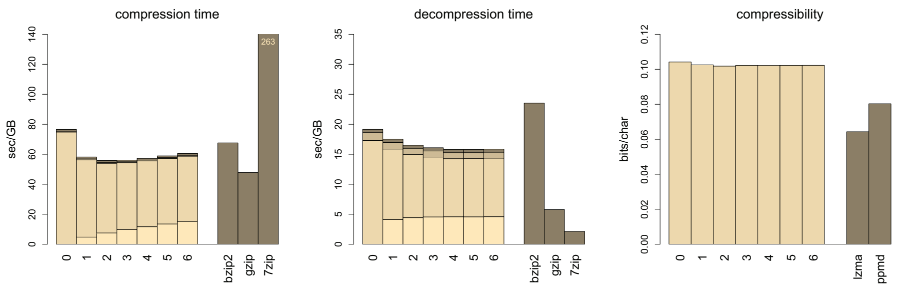
Wikipedia XML (enwik9), 1000MB, $\sigma = 206$



Part of human genome (dna), 404MB, $\sigma = 16$



36 versions of Linux kernel sources (kernel), 258MB, $\sigma = 160$



REFERENCES

[1] J. Kärkkäinen, D. Kempa, and S. J. Puglisi. Slashing the time for BWT inversion. In *Data Compression Conference*, pages 99–108. IEEE Computer Society, 2012.

[2] P. Mikkola. https://github.com/pjmikkol/bwtc, bwtc, May 2012. [9.5.2012].

[3] Y. Mori. http://code.google.com/p/libdivsufsort/, libdivsufsort, Nov. 2010. [9.5.2012].

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

SLASHING THE TIME FOR BWT INVERSION

Juha Kärkkäinen
Dominik Kempa
Simon J. Puglisi

The Burrows-Wheeler transform (BWT) is a powerful tool for data compression used for example in the popular bzip2 compressor. We describe new algorithms for inverting the BWT, which is a bottleneck in the decompression phase due to a high number of CPU cache misses.

One of the algorithms is consistently 2.3–4 times as fast as the previous state-of-the-art. Another algorithm achieves an asymptotic reduction in cache misses in theory and is the fastest algorithm in practice for highly repetitive data.

BURROWS–WHEELER TRANSFORM

The Burrows–Wheeler transform (BWT) is an invertible text transform defined as follows.

Input: text T = BANANA$

1. Build a matrix with the text *rotations* as rows

```
B A N A N A $
A N A N A $ B
N A N A $ B A
A N A $ B A N
N A $ B A N A
A $ B A N A N
$ B A N A N A
```

2. Sort the rows

$$\begin{array}{c|ccccc|c} F & & & & & & L \\ \hline \$ & B & A & N & A & N & A \\ A & \$ & B & A & N & A & N \\ A & N & A & \$ & B & A & N \\ A & N & A & N & A & \$ & B \\ B & A & N & A & N & A & \$ \\ N & A & \$ & B & A & N & A \\ N & A & N & A & \$ & B & A \end{array}$$

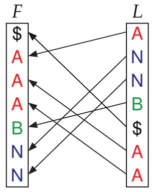
Output: BWT L = ANNB$AA (the last column)

The properties of the BWT make it easier to compress than the original text. It is used as the first stage in many compression programs including the widely used bzip2.

INVERSE BWT

Assume $L[i]$ is the k-th occurrence of a in L. We define $LF[i] = j$, where $F[j]$ is the k-th occurrence of a in F.

```
Algorithm SimpleInverse
1: ComputeLF
2: p ← locate(L, $)
3: for i ← 0 to n − 1 do
4:     T^R[i] ← L[p]
5:     p ← LF[p]
```



Computing LF is very fast, but the main loop suffers from multiple cache misses due to irregular memory access pattern hence it is slow in practice. Even with the optimized memory layout (see picture below) it can perform $\sim n$ cache misses. This algorithm is used in bzip2.

$LF[0]$	$L[0]$	$LF[1]$	$L[1]$	$\cdots$

FASTER ALGORITHMS FOR INVERSE BWT

Our algorithms can be divided into general purpose inversion algorithms (fast for all strings) and algorithms optimized for repetitive input.

SUPER ALPHABET TECHNIQUE

To reduce the number of cache misses in a general case we add a very fast (cache-friendly) preprocessing stage that allows restoring two characters at a time in the main inversion loop. We precompute for each position i:

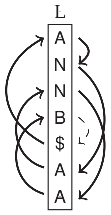
$$LF^2[i] = LF[LF[i]]$$
$$LL[i] = L[i]L[LF[i]]$$

The main loop of the inversion then becomes:

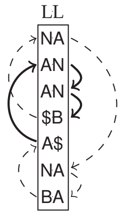
```
3: for i ← 0 to n/2 do
4:     T^R[2i..2i + 1] ← LL[p]
5:     p ← LF^2[p]
```

Assuming we use a similar memory layout as with SimpleInverse (LF^2 and LL stored interleaved) the number of cache misses is halved.

This is illustrated in the picture below (solid arcs represent the paths traversed in the main loop).



Byte-by-byte approach [3]
$\sim n$ cache misses



Super-alphabet [1]
$\sim n/2$ cache misses

MULTIPLE STARTING POSITIONS

To reduce the *cost* of cache misses we start the inversion from several positions simultaneously. Such computations are independent hence could be parallelized.

We use no explicit parallelism, but interleave the computations. Modern CPUs allow *out-of-order execution*: while one computation is waiting for a cache miss, others (independent) can proceed.

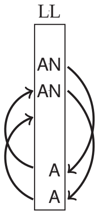
SPEEDUP FOR REPETITIVE INPUT

Strings containing lots of repeated factors offer a possibility of saving cache misses: once a frequent factor has been restored, other occurrences can be sequentially copied from that first one.

BWT captures repeating factors in the form of runs of equal symbols which affect LF mapping:

Lemma ([2]). *For any* $i \in 1..n-1$ *such that* $L[i] = L[i-1]$, $LF[i] = LF[i-1] + 1$.

Consequently, LF tends to contain lots of "parallel paths" (see example on the right). Such structure can be recognized from the BWT string and used to reduce cache misses.



The `copy` algorithm [2] detects local parallel paths in the main loop, halving the number of cache misses in the best case.

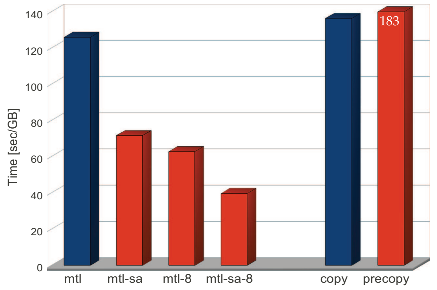
Our new algorithm called `precopy` [1] preprocess the data to detect more parallel paths and can reduce the asymptotic cache complexity.

EXPERIMENTAL RESULTS

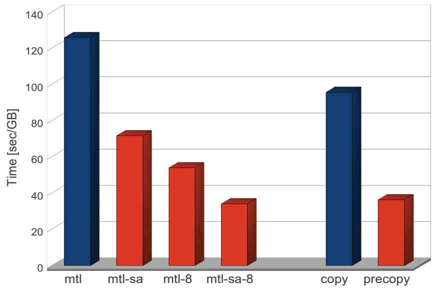
The graphs below show the runtime of the inversion algorithms (prior and new) on three files.

Name	Description
mtl	algorithm used in bzip2 [3]
mtl-sa	mtl with super-alphabet
mtl-8	mtl with 8 starting positions
mtl-sa-8	combination of preceding two
copy	local parallel path search [2]
precopy	precomputing parallel paths

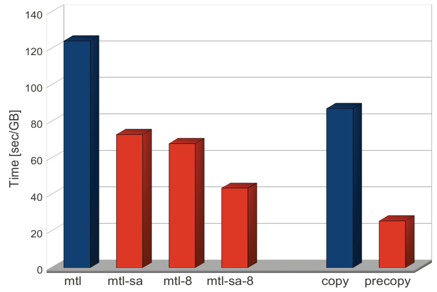
Part of human genome (100MiB)



100 x 1MiB english + mutations



36 versions of Linux kernel sources (246MiB)



REFERENCES

[1] J. Kärkkäinen, D. Kempa, and S. J. Puglisi. Slashing the time for BWT inversion. In *DCC*, pages 99–108. IEEE, 2012.

[2] J. Kärkkäinen and S. J. Puglisi. Cache friendly Burrows-Wheeler inversion. In *CCP*, pages 38–42. IEEE, 2011.

[3] J. Seward. Space-time tradeoffs in the inverse B-W transform. In *DCC*, pages 439–448. IEEE, 2001.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Indexing Finite Language Representation of Population Genotypes

Jouni Sirén, Niko Välimäki, Veli Mäkinen

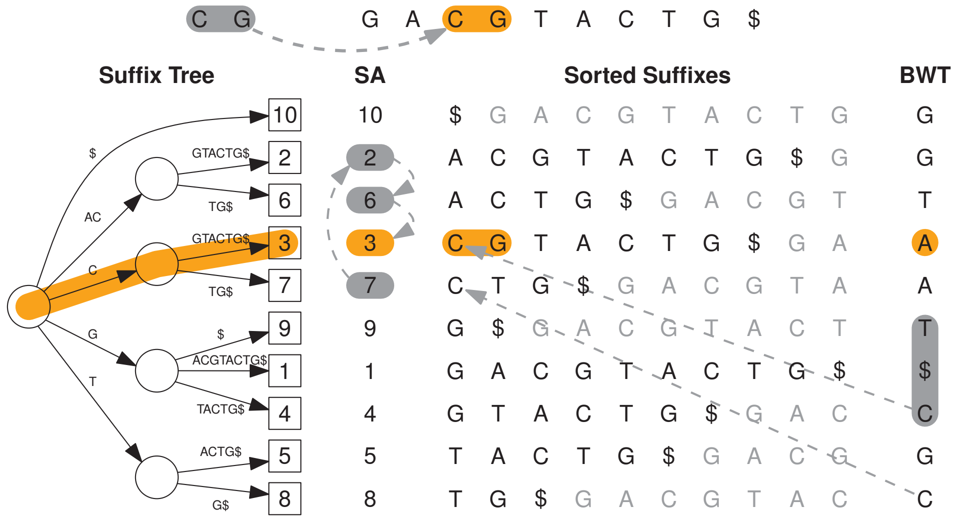
ABSTRACT

Compressed full-text indexes [6] based on the *Burrows-Wheeler transform (BWT)* are widely used in bioinformatics. Their most succesful application so far has been mapping short reads to a reference sequence (e.g. Bowtie [3], BWA [4], SOAP2 [5]). These indexes use the BWT to simulate the *suffix tree* or the *suffix array (SA)*, while using much less space than either of them. A simple generalization allows indexing a set of sequences.

We propose a biologically motivated generalization of the BWT to finite languages. Given a multiple alignment of sequences (e.g. individual genomes), we build a compressed index capable of simulating the suffix array over plausible recombinations of the sequences. Alternatively, we start from a reference sequence and a set of mutations, and build the index over sequences containing any subset of the mutations.

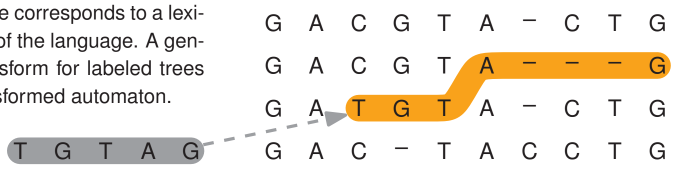
Our approach is based on finite automata. We start with an automaton recognizing the input language. This automaton is transformed into an equivalent automaton, where each state corresponds to a lexicographic range of suffixes of the language. A generalization of the XBW transform for labeled trees [2] is used to index the transformed automaton.

FULL-TEXT INDEXES FOR PATTERN MATCHING AND SEQUENCE ANALYSIS



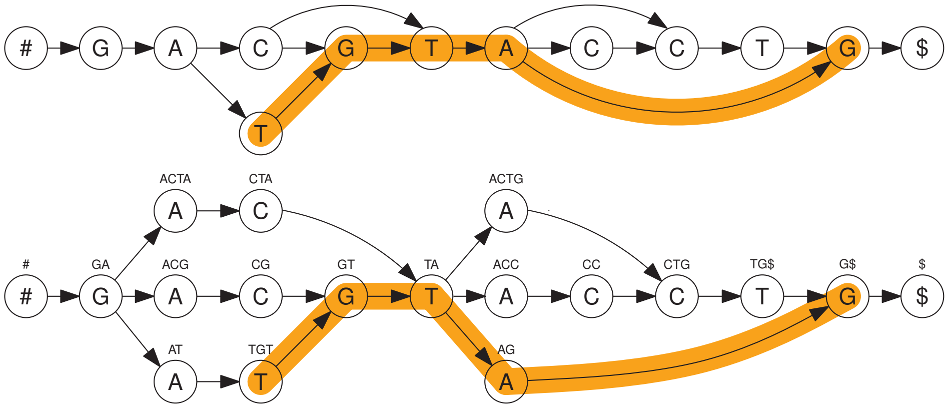
SA	Sorted Suffixes	BWT
10	$ G A C G T A C T G	G
2	A C G T A C T G $ G	G
6	A C T G $ G A C G T	T
3	C G T A C T G $ G A	A
7	C T G $ G A C G T A	A
9	G $ G A C G T A C T	T
1	G A C G T A C T G $	$
4	G T A C T G $ G A C	C
5	T A C T G $ G A C G	G
8	T G $ G A C G T A C	C

A MATCH IN MULTIPLE ALIGNMENT


T G T A G

G A C G T A – C T G
G A C G T A – – – G
G A T G T A – C T G
G A C – T A C C T G

INITIAL AUTOMATON AND SORTED AUTOMATON



GENERALIZED COMPRESSED SUFFIX ARRAY

	$	ACC	ACG	ACTA	ACTG	AG	AT	CC	CG	CTA	CTG	G$	GA	GT	TA	TG$	TGT	#
BWT	G	T	G	G	T	T	G	A	A	A	AC	AT	#	CT	CG	C	A	$
Edges	1	1	1	1	1	1	1	1	1	1	1	1	100	1	100	1	1	1

Basic operations are about 2 times slower than in regular BWT-based indexes. For reasonable mutation frequencies f, the expected size of the sorted automaton is $n(1+f)^{O(\log n)}$, where n is the length of the reference sequence. For $1/f = \Omega(\log n)$, this becomes $O(n)$. In our experiments, an index built for the human reference genome and the genetic variation found in the Finnish population sample of the *1000 Genomes Project* took approximately 2.8 gigabytes.

FUTURE DIRECTIONS

- With our current algorithm, the construction of a genome-scale index requires 12 hours and 192 gigabytes of memory. We are currently investigating other algorithms, such as external memory construction and distributed construction in the MapReduce framework [1].
- In principle, our index can be used in any algorithm using a regular BWT-based index. What can be done efficiently in practice?
- We are currently investigating several ways to use the generalized index in read alignment. Are there other applications, where our index could be superior to the existing approaches?

REFERENCES

[1] J. Dean, S. Ghemawat: *Simplified Data Processing on Large Clusters*. OSDI 2004.

[2] P. Ferragina et al.: *Compressing and indexing labeled trees, with applications*. Journal of the ACM, 2009.

[3] B. Langmead et al.: *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome*. Genome Biology, 2009.

[4] H. Li, R. Durbin: *Fast and accurate short read alignment with Burrows-Wheeler Transform*. Bioinformatics, 2009.

[5] R. Li et al.: *SOAP2: an improved ultrafast tool for short read alignment*. Bioinformatics, 2009.

[6] G. Navarro, V. Mäkinen: *Compressed full-text indexes*. ACM Computing Surveys, 2007.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

DISTRIBUTED STRING MINING ALGORITHM FOR HIGH-THROUGHPUT SEQUENCING DATA

Niko Välimäki

STRING MINING UNDER FREQUENCY CONSTRAINTS

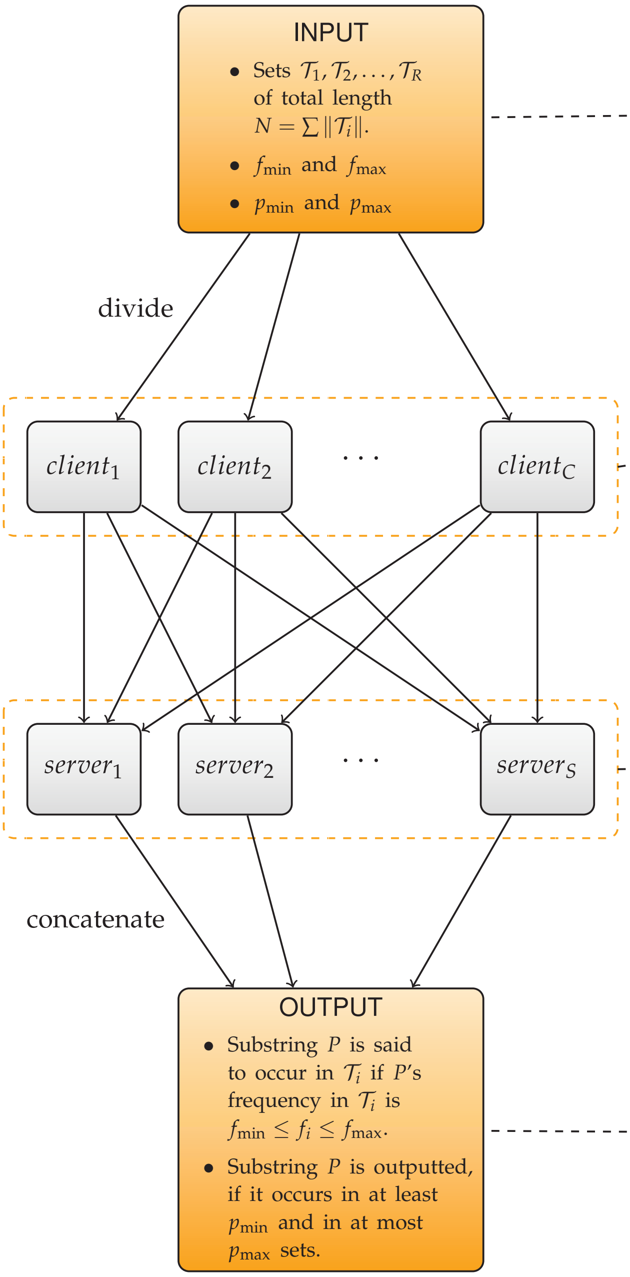
- The goal is to extract *emerging substrings* that discriminate two (or more) datasets.

$$\mathcal{T}^+ = \{\ \texttt{I am positive,}\ \texttt{I am also positive,}\ \texttt{I am also positive}\} \qquad \mathcal{T}^- = \{\ \texttt{I am negative,}\ \texttt{I am also negative,}\ \texttt{I am not negative}\}$$

- Substring `I am` is highly frequent but makes no difference.
- Substrings `positive` and `negative` clearly differentiate $\mathcal{T}^+$ from $\mathcal{T}^-$.

Method	Time	Space (in bits)
Fischer-Huen-Kramer'06	$\mathcal{O}(N)$	$\mathcal{O}(N \log N)$
Kügel-Ohlebusch'08	$\mathcal{O}(RN)$	$\mathcal{O}(\max_i \lVert\mathcal{T}_i\rVert \cdot \log N)$
Fischer-Mäkinen-Välimäki'08	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log \sigma + R \log N)$
Dhaliwal-Puglisi-Turpin'12	$\mathcal{O}(N \log^2 N)$	$\mathcal{O}(N \log \sigma + R \log N)$

- Existing methods are practical up to a few gigabytes of input.
- We introduce a distributed algorithm that requires less space than KO'08 per node and has a competitive time complexity.



INPUT

- Sets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_R$ of total length $N = \sum \lVert\mathcal{T}_i\rVert$.
- $f_{\min}$ and $f_{\max}$
- $p_{\min}$ and $p_{\max}$

OUTPUT

- Substring P is said to occur in $\mathcal{T}_i$ if P's frequency in $\mathcal{T}_i$ is $f_{\min} \le f_i \le f_{\max}$.
- Substring P is outputted, if it occurs in at least $p_{\min}$ and in at most $p_{\max}$ sets.

HUMAN GUT METAGENOMICS

# of datasets (R)	124 individuals
# of reads	2.8 billion
Read length (ℓ)	44–75 bases
Alphabet (σ)	{`A, C, G, T`}
Total size (N)	0.4 terabases

CLIENT SIDE PROCESSING

1. Simulate a suffix tree traversal via *suffix array* & *LCP array*.
2. Compute frequencies and check against $f_{\min}$ and $f_{\max}$.

	Worst-case	Expected
Time	$\mathcal{O}\left(\frac{N}{C}\ell \log N\right)$	$\mathcal{O}\left(\frac{N}{C} \log^2 N\right)$
Space (in bits)	$\mathcal{O}\left(\frac{N}{C} \log \sigma\right)$	$\mathcal{O}\left(\frac{N}{C} \log \sigma\right)$

In practice, about ten hours using $f_{\min} = 10$, $f_{\max} = \infty$, $C = 274$ and each client requiring ≈0.5 GB of main memory.

SERVER SIDE PROCESSING

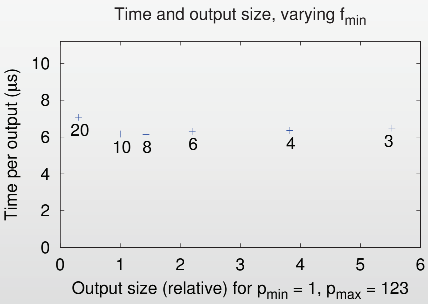
1. Merge the (sorted) input from clients on the fly.
2. Output substrings that obey the constraints $p_{\min}$ and $p_{\max}$.

	Worst-case	Expected
Time	$\mathcal{O}(N\ell)$	$\mathcal{O}\left(\frac{N}{S} \log N\right)$
Space (in bits)	$\mathcal{O}(C\ell \log N)$	*negligible*
Transmission bit-load	$\mathcal{O}\left(\frac{N}{CS}\ell\right)$	$\mathcal{O}\left(\frac{N}{CS} \log N\right)$

In practice, about ten hours using $S = 4$ servers for any $p_{\min}$, $p_{\max}$.

APPLICATIONS AND FEASIBILITY

- Sequence classification, knowledge discovery, comparative metagenomics
- Collaboration with Antti Honkela and Samuel Kaski's group.



ACKNOWLEDGEMENTS

Elisabeth Georgii, Antti Honkela, Samuel Kaski, Veli Mäkinen, Simon J. Puglisi, Leena Salmela, Sohan Seth

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

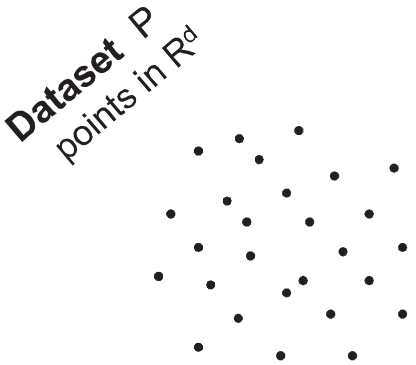
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Geometric Data Summarization Simplified and Improved

Dan Feldman
CSAIL
Department of Electrical Engineering and Computer Science
MIT

Juha-Antti Isojärvi
Helsinki Institute for Information Technology
Department of Mathematics and Statistics
Department of Computer Science
University of Helsinki

Valentin Polishchuk
ALGODAN
Helsinki Institute for Information Technology
Department of Computer Science
University of Helsinki



Measure m(P)
$m : 2^{R^d} \rightarrow R$

Convex hull volume

Radius of enclosing sphere

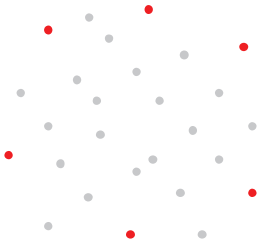
Bounding box dimensions

Diameter

Convex hull surface area

Directional width

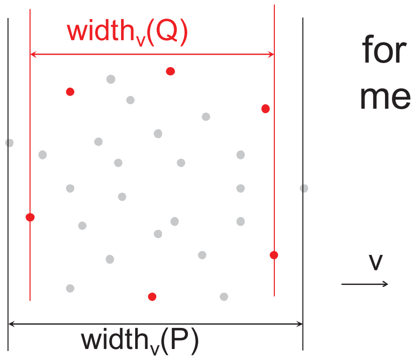
Coreset Q
subset of P



$m(Q) \sim (1 \pm \varepsilon)\, m(P)$

$|Q| = f(\varepsilon)$
independent of |P| !

ε-kernel : very powerful coreset



for many measures

Agarwal, Har-Peled, Varadarajan, "Approximating extent measures of points", J. ACM, 51(4), 2004

Involved construction and proofs

Approximates directional width
$| \text{width}_v(Q) - \text{width}_v(P) | \leq \varepsilon \bullet \text{witdh}_v(P)$

Our work: better ε-kernels

Simple construction and proofs

Better approximation of directional width in 2D
$| \text{width}_v(Q) - \text{width}_v(P) | \leq \varepsilon \bullet \min_v \text{witdh}_v(P)$

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

COMPRESSION-BASED CLUSTERING OF CHROMAGRAM DATA: NEW METHOD AND REPRESENTATIONS

Teppo E. Ahonen

teahonen@cs.helsinki.fi

ABSTRACT

We approach the problem of clustering chromagram data by presenting two new single-dimensional representations and using a compression-based distance metric for the k-medians clustering process. The method is evaluated using real-world audio cover version data.

CLUSTERING

We cluster the chromagram data with a modified version of the well-known k-medians algorithm. The distances between the strings are calculated using normalized compression distance (NCD) [1]. In the clustering phase, the new cluster centroid median is selected according to length-increasing, lexicographical order of the strings.

For two strings *x* and *y* NCD is denoted

$$NCD(x,y)=\frac{C(xy)-min\{C(x),C(y)\}}{max\{C(x),C(y)\}}$$

where *C(x)* is the length of the string *x* when compressed using compression algorithm *C*, and *xy* is the concatenation of *x* and *y*.

For compression, we use the bzip2 algorithm.

OPTIMAL TRANSPOSITION INDEX (OTI)

OTI [2] is the value of the most likely semitone transposition between two chromagrams. For two global chroma profiles (chromagrams summed over time and normalized) G_a and G_b, the OTI is denoted

$$OTI(G_a,G_b)=arg\,max\{G_a\cdot circshift(G_b,i-1),1\leq i\leq 12\}$$

REPRESENTATIONS

We apply OTI to produce a sequnce of characters from the chromagram data. For each chroma frame, we calculate the OTI value between the frame and the global chromagram of the piece, resulting to a sequence of values from 0 to 11. For the lack of a better term, we call this **chroma contour**. Formally, for a chromagram g_a of length *i* and its global chroma profile G_a, the chroma contour sequence is

$$ccs(i)=OTI(g_a(i),G_a)$$

The representation has the advantage of being key-invariant. However, when comparing two pieces of music, it would seem fruitful to use their similarities already when processing the sequences. Here, we apply OTI to the chromagram of the target and the global chroma profile of the query. Again, for the lack of a better term, we call this **cross-chroma contour**. Formally, for a target chromagram g_a of length *i* and a query global chroma profile G_b, the cross-chroma contour sequence is

$$cccs(i)=OTI(g_a(i),G_b)$$

The cross-chroma contour is not key invariant. In order to transpose two chroma sequences to the common key, we apply OTI to their global chromagrams and transpose the query according to the OTI value before producing the cross-chroma sequence.

As the sequences produced by the method seemed to oscillate rapidly between values, we experimented reducing the noise of the data by using median filtering. The filtering was applied to both chromagram data and the contour sequences. However, based on the evaluation results, it seems that the noisy sequences actually produce higher results, suggesting that the noise contains distinguishing information.

EVALUATIONS

In order to validate the performance of our system, we constructed a dataset of 10 cover versions of 12 pieces of music, thus totaling 120 pieces of music. We experimented with subsets of 30, 60, and 120 pieces, with k values of 3, 6, and 12, respectively.

The clustering performance was measured using cluster purity, and as the k-medians algorithm selects the initial cluster centroids randomly, the tests were run five times, and the averaged results are reported here.

METHODS	SET30	SET60	SET120
CHROMA CONTOUR	0.367	0.283	0.217
CROSS-CHROMA CONTOUR	0.374	0.317	0.257
CC + CHROMA FILTERING	0.310	0.231	0.162
CCC + CHROMA FILTERING	0.344	0.312	0.228
CC + SEQ. FILTERING	0.331	0.258	0.189
CCC + SEQ. FILTERING	0.337	0.294	0.212
CC + BOTH FILTERINGS	0.133	0.104	0.081
CCC + BOTH FILTERINGS	0.192	0.162	0.132
RANDOM BASELINE	0.233	0.117	0.067

CONCLUSIONS

The proposed method has potential for chromagram clustering. Using cross-chroma contour provides slightly higher accuracy than chroma contour, and processing data too much causes over-simplification. The method seems robust, as increase in the data does not result in significantly worse results.

REFERENCES

[1] R. Cilibrasi and P. M. B. Vitanyi: Clustering by Compression. IEEE Trans. Information Theory 51:4(2005)

[2] J. Serra, E. Gomez and P. Herrara: Transposing Chroma Representations to a Common Key. IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects 2008

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

DISTINGUISHING BETWEEN MAJOR AND MINOR CHORDS IN AUTOMATIC CHORD TRANSCRIPTION

Antti Laaksonen

Automatic chord transcription is a problem of extracting the harmonic content from a music signal and representing it through chord symbols. We focus on distinguishing between major and minor chords in automatic chord transcription. We are especially interested in the role of the musical context in this process. We conduct an experiment where human listeners are asked to classify chords which a computer transcriber has failed to recognize when evaluated using a collection of Beatles songs. Based on this experiment and our analysis, we conclude that the musical context is often needed in distinguishing between major and minor chords. Furthermore, sometimes the quality of a chord cannot be unambiguously determined even if the full musical context is available.

BACKGROUND

CURRENT SOLUTIONS

Automatic chord transcribers are usually combinations of low-level signal processing methods and high-level probabilistic models.

The most popular evaluation dataset for automatic transcribers has been the Beatles dataset [5] which offers hand-made reference chord annotations for a collection of Beatles songs.

The best automatic transcribes have achieved a transcription rate of about 80% in MIREX chord transcription task [4]. One of them is Mauch's MM1 [3] which is purely based on the audio data without using the musical context.

We collected a set of 454 audio segments from the Beatles material where the chord proposed by MM1 differs from the ground truth.

- In 202 segments (45%), a major chord was recognized as a minor chord or vice versa.
- In 93 segments (20%), there are problems with chord alignment or tuning, or there is no clear chord content.
- In 159 segments (35%), there is a meaningful chord in the ground truth, but the proposed chord is different in some other way than in the first class.

MUSICAL CONTEXT

The melody, the chords and the key of a musical piece are connected with each other. For example, if the melody of a piece is known, there are usually only a couple of typical chords to choose from. These factors affecting the probabilities of chords are called the musical context.

Important parts of the musical context are:

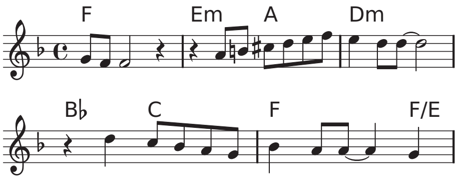
- **Chord transitions**: a G major chord is often followed by a C major chord. An E major chord would be a big surprise.
- **Key**: Typical chords in C major key are C, F, and G majors and D, E, and A minors.
- **Structure**: Especially in popular music, there are repeating chord sequences.
- **Melody**: The melody significantly limits the set of possible chord sequences.

While there are several ways to estimate chord probabilities in a musical context [1, 2, 6], it is not clear how important the probabilities are in chord transcription. Professional human transcribers also make guesses but they never publish transcriptions that do not sound good. The reason for this is that a guess is always followed by verification: if the guessed chord does not sound good, it is simply rejected.

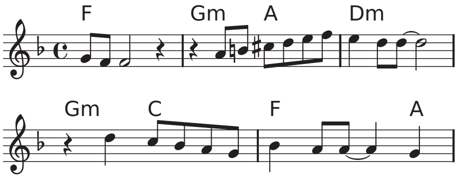
ALTERNATIVE TRANSCRIPTIONS

There are often several good chord transcriptions for a musical piece. Consider the following two transcriptions of the Beatles' *Yesterday*.

First transcription:



Second transcription:



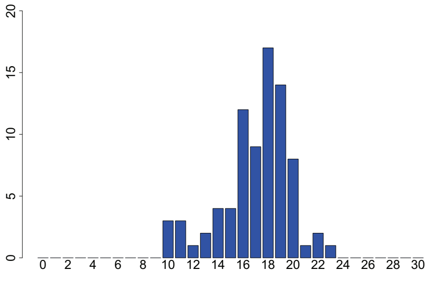
The first transcription is a rather accurate transcription from the original studio album, while the second transcription contains three different chords. However, both the transcriptions sound good and an average listener can hardly notice any difference between them. It would be misleading to state that the first transcription is ”correct” and the second one is ”incorrect”.

MAJOR AND MINOR CHORDS

EXPERIMENT

We conducted an experiment to study the ability of human listeners to distinguish between major and minor chords without the musical context. A total of 81 people with a musical background participated in our experiment.

The experiment consisted of 30 audio segments randomly selected from our collection. At each segment, the participants were asked to determine whether the chord is major or minor. The following diagram shows the number of correct answers. Surprisingly, there is only a slight improvement over a totally random choice.



ERROR GROUPS

We found out that the segments we examined can be divided into three groups:

- **Easy chords** which almost all participants recognized correctly. An usual problem on the signal processing level is that even in a pure minor chord, the fifth harmonic of the root note is a major third.
- **Unclear chords** which contain both major and minor elements. One interpretation is that the third in the accompaniment determines the chord, but the problem is that the third in the melody is often played more strongly.
- **Erroneous chords** where we disagree with the ground truth. The reference chords should not be used without caution. Of course, the quality of a chord is often a subjective decision.

The first two groups cover most of the segments we examined. In the experiment, there were four segments that fall in to the third group.

CONCLUSION

So far, automatic chord transcribers have been evaluated using a ground truth with a single reference chord for each audio segment. However, even in distinguishing between major and minor chords, there are often valid arguments for both interpretations. This suggests that the traditional goal to maximize the number of chords matching the ground truth only partially captures the properties of a good chord transcription.

REFERENCES

[1] J. Bello and J. Pickens: ”A Robust Mid-level Representation for Harmonic Content in Music Signals,” *Proc. ISMIR 2005*

[2] M. Mauch and S. Dixon: ”Approximate Note Transcription for the Improved Identification of Difficult Chords,” *Proc. ISMIR 2010*

[3] M. Mauch: ”Simple Chord Estimate: Submission to the MIREX Chord Estimation Task,” 2010

[4] http://www.music-ir.org/mirex/wiki/

[5] http://isophonics.net/content/reference-annotation-beatles

[6] M. Ryynanen and A. Klapuri: ”Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music,” *Computer Music Journal*, 32(3), 2008

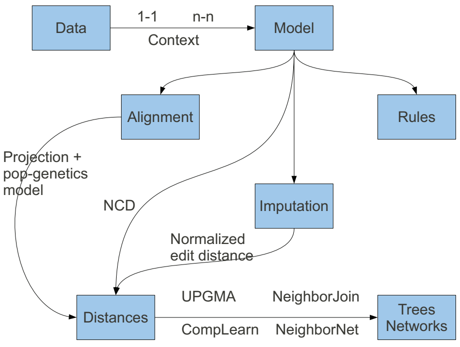
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
Faculty of Science
Department of Computer Science
ALGODAN Center of Excellence

Hannes Wettig
Suvi Hiltunen
Roman Yangarber

Analysis of Etymological Data via MDL

We develop MDL-based models for studying etymological data. The data consists of *cognate sets*: sets of genetically related words—words deriving from a common (unobserved) ancestor in the proto-language—in different (observed) languages within a language family. One goal is to find the best possible *alignment* of all the words in the data. The alignment must respect the *Principle of Regular Sound Correspondence*: sound changes that occur as a given language evolves are not random, but apply deterministically throughout the language, typically conditioned on the features and the context of the sound. Thus, a complementary goal is to discover the rules of sound change that best describe the data.

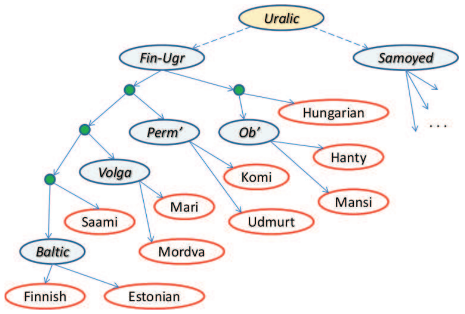
OUTLINE



DATA

We have several databases of *cognate sets* from different language families, including the Uralic family. The databases may conflict with regard to inclusion of specific words in a cognate set.

ID	EST	FIN	KHN	KOM	MAN	MAR	MRD	SAA	UDM	UGR
91	-	-	-	ćepeĺ ?	-	cəwešte ?	-	-	čepiĺt ?	csip ?
92	-	-	-	-	-	čeve ?	-	-	-	csëpp ?
93	-	-	-	ťovťa ?	śüĺś ?	-	-	-	-	-
94	-	-	-	śuź ?	-	-	-	cisku ?	-	sas ?
95	-	-	-	ǯoǯ	-	-	-	-	ǯiǯol	-
96	-	-	čăčə	ćuž	šošəɣ	šača	šačo	-	-	-
97	ammak	hama	čăma	-	šoməɣ	-	-	-	-	-
98	-	-	-	-	-	-	-	cuoʒ'ʒå	-	-
99	-	-	čuš	-	šuš	-	šašto	-	-	-
100	-	-	čoŋχ	-	šaŋk	čaŋɣe	čavo	-	-	-
101	-	-	-	-	-	šapka ?	-	-	-	sápad ?
102	hape ?	hapan ?	-	-	-	šapə̑	čapamo ?	-	-	savanyú
103	-	-	čȧkən	ǯagal	šā̊kət	-	-	-	ǯ́okal	čäk
104	händ	häntä	čĕṇč	-	šįs	-	-	-	-	-



THE OBJECTIVE

We begin with pairwise alignment—one language pair at a time.

According to the Minimum Description Length (MDL) principle, we can compress the data effectively if we can discover *regularity* in the data. This regularity is the laws of sound change that we seek.

Thus, the objective function that we optimize is the MDL codelength; using Bayesian marginal likelihood, or *prequential* coding:

$$L(D) = -\sum_{e\in E}\log\Gamma\big(c(e)+\alpha(e)\big) + \sum_{e\in E}\log\Gamma\big(\alpha(e)\big) + \log\Gamma\Big[\sum_{e\in E}\big(c(e)+\alpha(e)\big)\Big] - \log\Gamma\Big[\sum_{e\in E}\alpha(e)\Big]$$

Using Normalised Maximum Likelihood (NML) gives somewhat better compression overall.

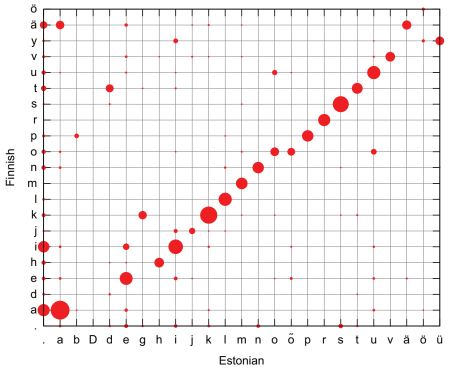
BASELINE (1-1) ALIGNMENT MODEL

For a given word-pair, many alignments are possible: Finnish and Hanty words meaning *year*:

v	*u*	*o*	*s*	*i*		*v*	*u*	*o*	*s*	*i*	
|	|	|	|	|		|	|	|	|	|	etc...
a	*l*	.	.	.		.	*a*	.	*l*	.	

(The symbol "**.**" indicates deletion or insertion.)

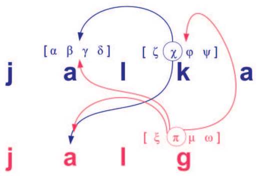
Search algorithm: begin with a random alignment, and iteratively realign one word pair at a time via Dynamic Programming, using the currently best alignment of the remainder of the data.



The algorithm converges to a (locally optimal) alignment of the complete data. The area of the circle is proportional to the probability mass of each 1-1 symbol alignment.

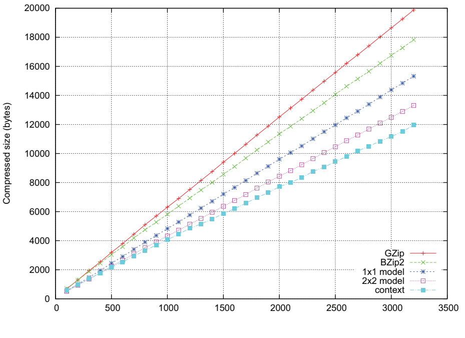
CONTEXT MODELS

We code each sound σ as a vector of phonetic features, **and** coding is conditioned on (features of) sounds in the context of σ—the model can query the history that has been coded so far.



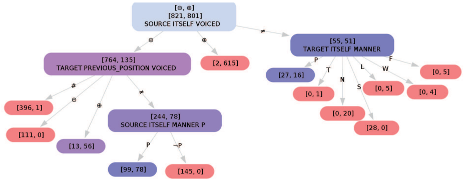
COMPRESSION RATES

The test of the model "goodness" is compression power: the cost of the complete (aligned) data:



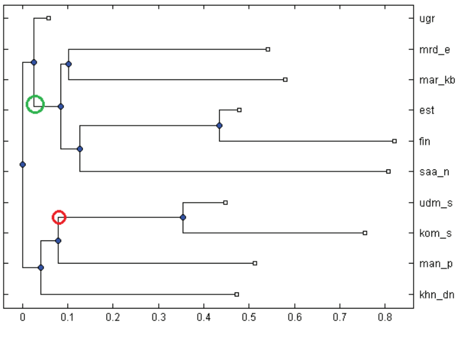
RULES AS DECISION TREES

The model learns one tree for coding each feature of a sound, minimizing the tree cost. Each node queries the history to help prediction.



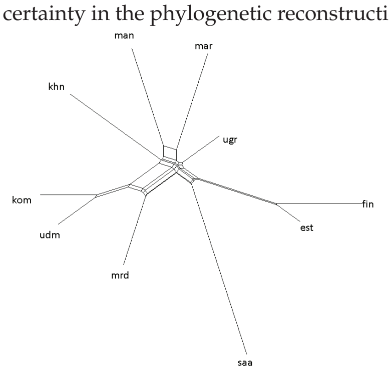
RECONSTRUCTING PHYLOGENIES

We obtain pairwise language distances in several ways from the alignment models, and induce trees using, e.g., UPGMA, NeighborJoin:

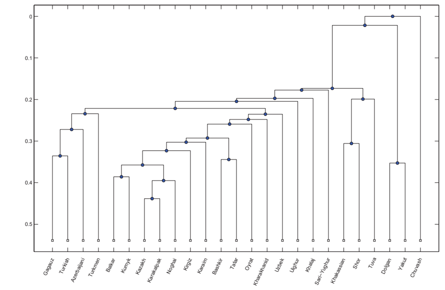


PHYLOGENETIC NETWORKS

NeighborNet (SplitsTree) helps identify the uncertainty in the phylogenetic reconstructions:



Applying to other language families: Turkic



Analysis of Linguistic Variation

Jefrey Lijffijt

Aalto University, Finland

Abstract

- Many medium to large text corpora have been compiled and annotated
- This enables the study of more diverse and detailed aspects of language
  - E.g., differences between writing style of various age groups/gender/media
- New computational and statistical challenges arise

Burstiness

- In linguistics it is often assumed that all words in a corpus are independent
- It has been argued that this is not problematic when there are many samples
- Figure 1 shows how false this statement is
- This effect is known as *burstiness* [2]

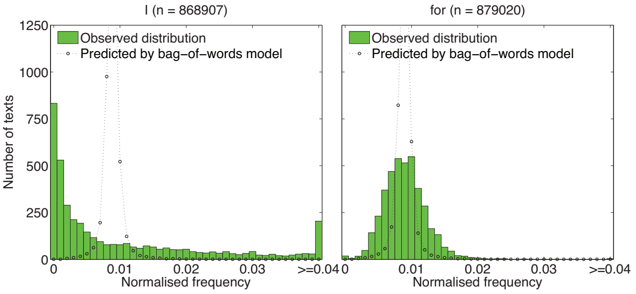


Figure 1: Frequency histograms of the words *I* and *for* in the British National Corpus. The distributions predicted under the bag-of-words assumption are very poor. The pronoun *I* is much burstier than the grammatical word *for* ; the Weibull shape parameter β is 0.57 and 0.93, for *I* and *for* respectively, see the paragraph below. Adapted figure from [4].

Inter-arrival times

- An *inter-arrival time* of a word is the number of words between two consecutive occurrences

" Finnair believes that it will be able to resume its scheduled service to **and** from New York on Monday, after two days of cancellations caused by hurricane Irene. All three airports serving New York City have been closed because of the hurricane **and** Finnair was forced to cancel flights on Saturday **and** Sunday. The airline is not certain when its scheduled service can be resumed, but the assumption is that Monday afternoon's flight from Helsinki will depart. Some Finnair passengers whose final destination is not New York have been rerouted **and** some have delayed travel plans. The company has also offered ticket holders a refund. *YLE* "

- $\mathrm{IAT}_{\mathrm{and}} = \{29, 9, 39, 29\}$

- The distribution of inter-arrival times describes the burstiness of a word
- A summary is obtained by fitting a Weibull distribution [1]

Comparison of word frequencies

- We can use statistical testing to find significant variations in writing styles
  - I.e., between time periods, between people or between text types
- Tests commonly employed are based on the bag-of-words assumption (χ^2-test)
- *Burstiness* leads to over-estimation of the significance [4]
- Improved tests based on inter-arrival times or bootstrapping are proposed [4]

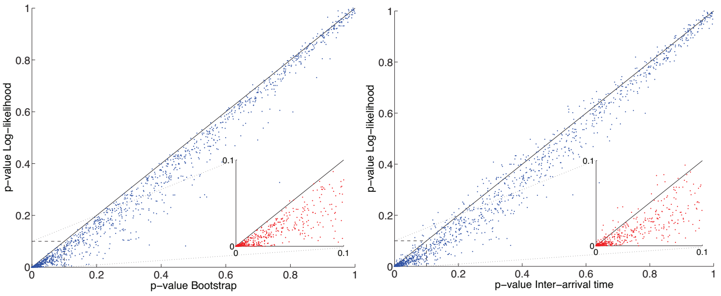


Figure 2: Comparison of p-values for the null hypothesis that the word is equally frequent in the two periods (1600-1639 and 1640-1681) of the Parsed Corpus of Early English Correspondence, for all words in the corpus. Both the bootstrap and inter-arrival time tests are often much more conservative than the log-likelihood ratio test.

Profile

- Doctoral student in the group of Heikki Mannila
- Member of ALGODAN, HIIT, PASCAL2
- Research interests include analysis of sequential data and mining bursty patterns
- E-mail: jefrey.lijffijt@aalto.fi

Classification of text genres

- Models for genre classification are complex and difficult to interpret
- It appears the main genres (of British English) can be recognized using a simple model and easy to compute surface level features [3]

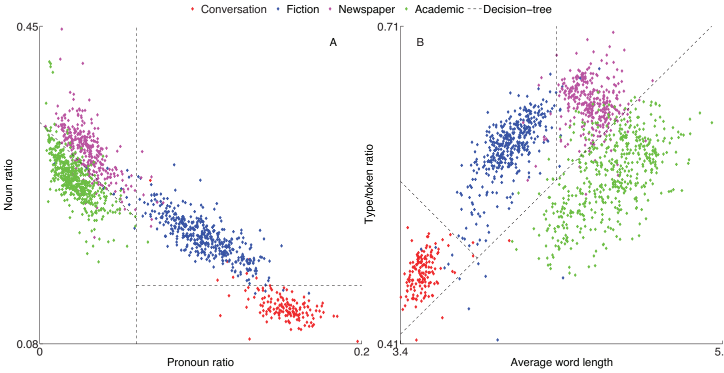


Figure 3: Two models for classification of the main genres of British English. The model was trained using the C4.5 algorithm on the British National Corpus, using both the original features and their cross-terms. Figure taken from [3].

Learning complex queries

- Linguists would often like to query a corpus for complex constructs
  - For example, *premodifying -ing participles* [5]
  - These are -ing participles that modify a noun, e.g., 'the *barking* dog'
- Straightforward queries have low recall because parsers and part-of-speech taggers are imperfect
- A query is essentially the same as a binary classifier
- We can *learn* complex queries just like training a classifier

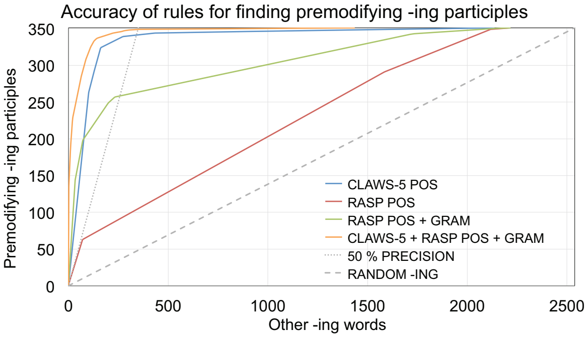


Figure 4: Precision and recall for classifiers based on several sources of information, based on a sample of 2902 *-ing* words, of which 351 are premodifying *-ing* participles, from the British National Corpus. Figure taken from [5].

References

[1] Altmann, Pierrehumbert & Motter 2009. "Beyond word frequency: bursts, lulls, and scaling in the temporal distributions of words". *PLoS One*, 4 (11): e7678.

[2] Katz 1996. "Distribution of content words and phrases in text and language modelling". *Natural Language Engineering*, 2 (1): 15–59.

[3] Lijffijt, Nevalainen & Mannila (*submitted*). "A simple model for recognizing core genres in the BNC".

[4] Lijffijt, Papapetrou, Puolamäki & Mannila 2011. "Analyzing word frequencies in large text corpora using inter-arrival times and bootstrapping". In *Proceedings of the ECML-PKDD 2011*.

[5] Vartiainen & Lijffijt 2012. "Premodifying -ing participles in the parsed BNC". In *Corpus Linguistics and Variation in English: Theory and Description.* Amsterdam/New York: Rodopi.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

ENSEMBLE COMPUTATION WITH OR- AND SUM-CIRCUITS

Matti Järvisalo, Petteri Kaski,
Mikko Koivisto and Janne H. Korhonen

ENSEMBLE COMPUTATION

An *ensemble* consists of a set of variables P and a family $\mathcal{Q}$ of subsets of P. For example:

$$P = \{x_1, x_2, x_3, x_4, x_5\}$$

$$\mathcal{Q} = \{\{x_1, x_2\}, \{x_1, x_2, x_3\}, \{x_1, x_4\}, \{x_1, x_4, x_5\}, \{x_1, x_2, x_3, x_4, x_5\}\}$$

The task is to compute either OR or SUM of variables in each set in $\mathcal{Q}$ using an *arithmetic circuit*.

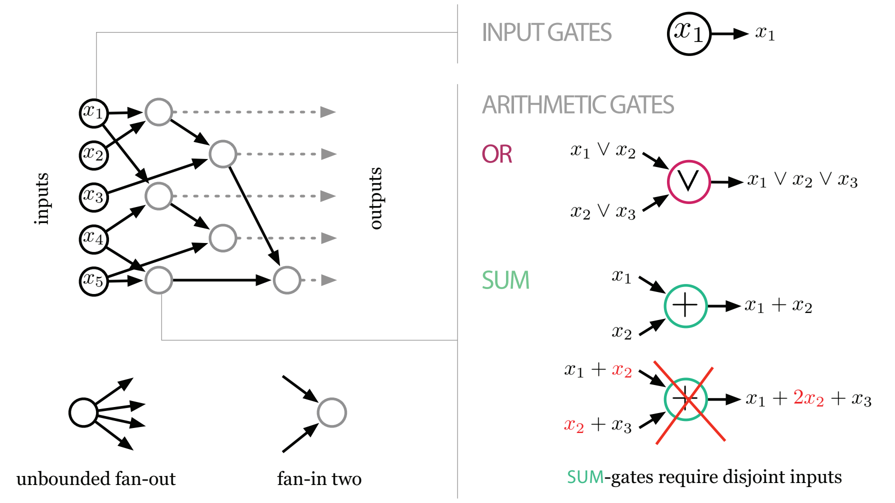
OR (Boolean variables)

$$\left\{\begin{array}{l} x_1 \vee x_2 \\ x_1 \vee x_2 \vee x_3 \\ x_1 \vee x_4 \\ x_1 \vee x_4 \vee x_5 \\ x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \end{array}\right.$$

SUM (natural numbers)

$$\left\{\begin{array}{l} x_1 + x_2 \\ x_1 + x_2 + x_3 \\ x_1 + x_4 \\ x_1 + x_4 + x_5 \\ x_1 + x_2 + x_3 + x_4 + x_5 \end{array}\right.$$

ARITHMETIC CIRCUITS



INPUT GATES $x_1 \rightarrow x_1$

ARITHMETIC GATES

OR: $x_1 \vee x_2$, $x_2 \vee x_3$ $\rightarrow$ $\vee$ $\rightarrow$ $x_1 \vee x_2 \vee x_3$

SUM: x_1, x_2 $\rightarrow$ $+$ $\rightarrow$ $x_1 + x_2$

$x_1 + x_2$, $x_2 + x_3$ $\rightarrow$ $+$ $\rightarrow$ $x_1 + 2x_2 + x_3$

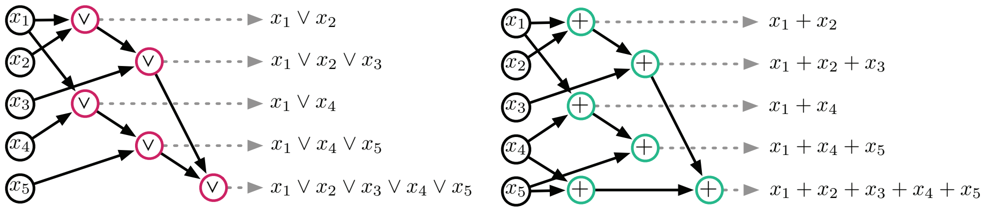
unbounded fan-out

fan-in two

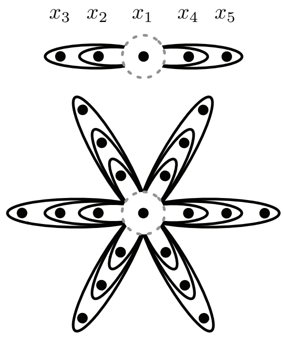
SUM-gates require disjoint inputs

SEPARATING OR- AND SUM-ENSEMBLES

There are ensembles for which the optimal OR-circuit has less gates that the optimal SUM-circuit. Our construction gives ensembles for which the SUM-circuit requires almost twice as many arithmetic gates. Finding better upper and lower bounds for the separation is an open question.

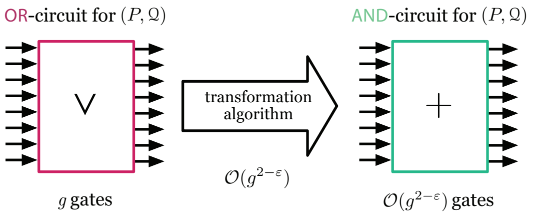


5 arithmetic gates vs. 6 arithmetic gates



generalising the construction

CIRCUITS AND EXPONENTIAL ALGORITHMS

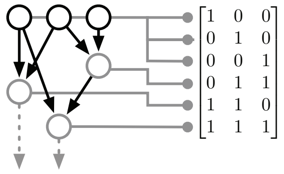


OR-circuit for $(P, \mathcal{Q})$
$\vee$
g gates
transformation algorithm
$\mathcal{O}(g^{2-\varepsilon})$
AND-circuit for $(P, \mathcal{Q})$
$+$
$\mathcal{O}(g^{2-\varepsilon})$ gates

Ensemble computation instances often arise in the context of exact exponential-time algorithms. In these cases, small OR-circuits are easy to find, but small SUM-circuits remain elusive.

In particular, existence of a sub-quadratic algorithm that transforms a given OR-circuit into a SUM-circuit for the same ensemble would violate the *Strong Exponential Time Hypothesis* and give improved algorithms for many NP-hard problems such as CNF-SAT.

SAT ENCODING AND FINDING CIRCUITS



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\bigwedge_{i=1}^{p} \Big(M_{i,i} \wedge \bigwedge_{j \neq i} \neg M_{i,j} \Big)$$

$$\bigwedge_{i=p+1}^{g} \bigvee_{k=1}^{i-2} \bigvee_{\ell=k+1}^{i-1} \bigwedge_{j=1}^{p} \Big((M_{k,j} \vee M_{\ell,j}) \leftrightarrow M_{i,j} \Big)$$

$$\bigwedge_{A \in \mathcal{Q}} \bigvee_{i=p+1}^{g} \Big[\Big(\bigwedge_{j \in A} M_{i,j} \Big) \wedge \Big(\bigwedge_{j \notin A} \neg M_{i,j} \Big) \Big]$$

An OR-circuit of size g exists for ensemble $(P, \mathcal{Q})$ if and only if there is a $|P| \times g$ binary matrix M that satisfies the formulas above. A similar encoding works for SUM-circuits.

We have used this encoding along with state-of-the-art Boolean satisfiability solvers to find the optimal circuits for all small non-isomorphic ensembles. Processing 1,434,897 ensembles took about 4 months of processor time. We did not find any larger separations between OR- and SUM-circuits than in the example above.

FOR MORE INFORMATION, see our paper: M. Järvisalo, P. Kaski, M. Koivisto, J. H. Korhonen. *Finding Efficient Circuits for Ensemble Computation*. SAT 2012

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

PARTIAL ORDER MCMC FOR STRUCTURE DISCOVERY IN BAYESIAN NETWORKS

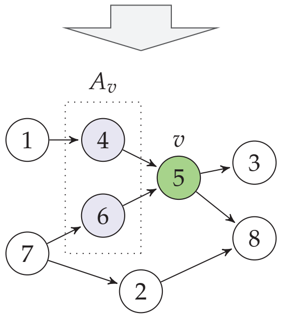
Teppo Niinimäki, Pekka Parviainen, Mikko Koivisto

We present a new Markov chain Monte Carlo method for estimating posterior probabilities of structural features in Bayesian networks. The method samples partial orders on the nodes; for each sample, the conditional probabilities of interest are computed exactly. Compared to previous methods our algorithm obtains a significant reduction in the size of sample space with negligible increase in computation time.

PROBLEM AND MOTIVATION

Learning the **structure** A of a **Bayesian network** from given **data** D is a problem that arises from the need to understand the dependencies or possible causality relations between the variables corresponding to the nodes of A.

	variables							
sample	1	2	3	4	5	6	7	8
1	2	1	0	1	2	2	2	1
2	2	0	2	2	0	2	2	0
3	2	0	1	1	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5000	2	2	1	2	0	2	0	1



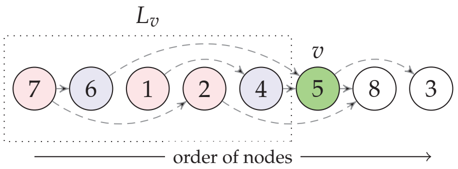
Instead of finding the most probable network structure (MAP) we want to compute the **posterior probability of each arc** by averaging over all structures.

NOTATION AND ASSUMPTIONS

The goal is to compute $p(f|D)$, where $f(A)$ is a binary **feature** function of interest, for example indicating whether a structure A contains given arc or not.

Structure prior:

For computational efficiency we assume an **order-modular** structure prior: The joint prior probability $p(A, L)$ of the structure A and a *linear order* L on the nodes factorizes to a product of local prior probabilities $\rho_v(L_v)q_v(A_v)$ over the nodes.



Similarly, the feature $f(A)$ is assumed to be a product of local features $f_v(A_v)$.

In addition we **limit the sizes of parent sets** $|A_v|$ to be at most k.

Why sampling?

Known exact methods scale up to about 30 nodes. For larger instances sampling based approximation is a natural choice.

METHOD

The state of the art methods are based on sampling linear orders of nodes by MCMC (Friedman and Koller, 2003). The resulting time requirement $O(n^{k+1})$ per sample is proportional to the number of possible parent sets.

The **general algorithm** is as follows:

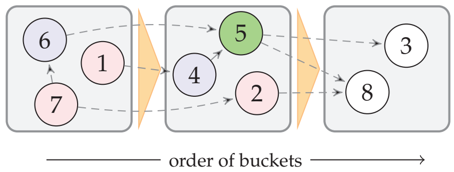
1. Sample orders $L_1, \ldots, L_T$ from $p(L|D)$.
2. Estimate $p(f|D) \approx \frac{1}{T}\sum_{i=1}^{T} p(f|D, L_i)$.

Instead of linear orders L **we suggest** sampling **partial orders** P on nodes of which sampling of linear orders is a special case. This has two consequences:

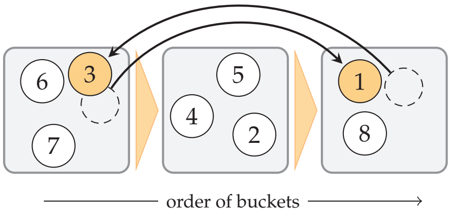
- The sample space can become significantly smaller as a single partial order sample usually corresponds to multiple linear orders samples. This can lead to **better mixing** in MCMC.
- The time complexity per sample becomes $O(n^{k+1} + n^2|\mathcal{I}(P)|)$ where $|\mathcal{I}(P)|$ is the number of *ideals* of P (Parviainen and Koivisto, 2010). For "thin" partial orders the first term dominates and the **increase of** the computational **cost is negligible**.

BUCKET ORDERS

As partial orders we use **bucket orders**.

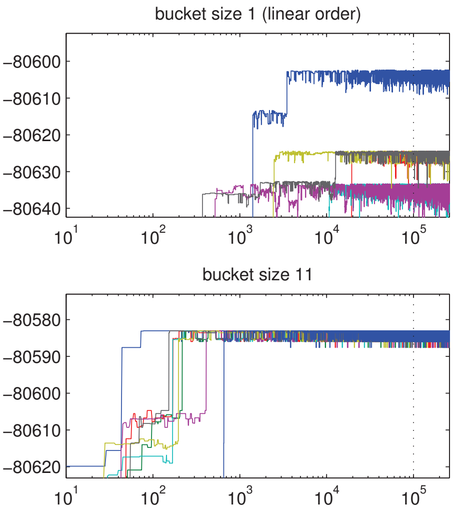


Sampling is based on **Metropolis–Hastings** MCMC algorithm with **swaps** of nodes between buckets as transitions.



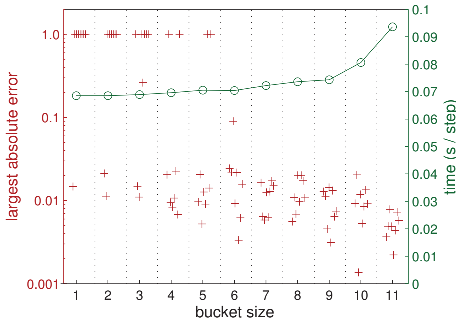
CONVERGENCE

The convergence of log-probability for MUSHROOM-dataset (8 independent runs):



ACCURACY AND TIME CONSUMPTION

The worst-case accuracy of estimates (8 independent runs) and time consumption for different bucket sizes:



HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

ANCESTOR RELATIONS IN THE PRESENCE OF UNOBSERVED VARIABLES

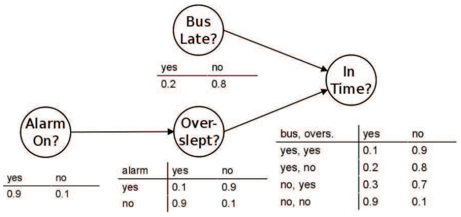
Pekka Parviainen, Mikko Koivisto

We present an exact dynamic programming algorithm for computing posterior probabilities ancestor relations, that is, directed paths in Bayesian networks. Our experimental results show that ancestor relations can be learned with good power even when a majority of involved variables are unobserved.

BAYESIAN NETWORKS

A Bayesian networks consists of two parts:

- The structure is a directed acyclic graph (DAG) that represents conditional independencies between variables.
- The parameters specify local probability distributions.



yes	no
0.2	0.8

yes	no
0.9	0.1

alarm	yes	no
yes	0.1	0.9
no	0.9	0.1

bus, overs.	yes	no
yes, yes	0.1	0.9
yes, no	0.2	0.8
no, yes	0.3	0.7
no, no	0.9	0.1

Compact, flexible and interpretable representations of a joint probability distribution.

Sometimes arcs are interpreted as cause-effect pairs.

STRUCTURE DISCOVERY

Construct the DAG from observational data.

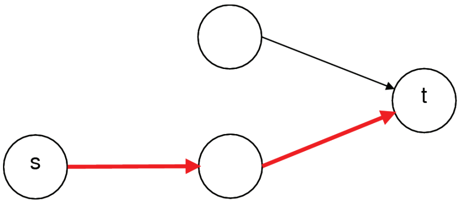
Challenges:

- The set of conditional independencies can be represented by a number of different DAGs (Markov equivalence class).
- There may be unobserved variables.
- Computational complexity.

ANCESTOR RELATIONS

There may be several almost equally good DAGs (or Markov equivalence classes) and the optimal DAG may be highly unlikely. Therefore, instead of learning an optimal DAG, it may be useful report probabilities of some *structural features* of interest, e.g., arcs.

Node s is an ancestor of node t in a DAG if there is a directed path from s to t in the DAG in question.



Ancestor relations are interpreted as (direct or indirect) causal relations.

RESEARCH QUESTIONS

Can ancestor relations be learned reliably if there are some unobserved variables at work?

Does learning ancestor relations yield more information than learning arcs?

Can ancestor relations be learned significantly faster than by a brute force algorithm?

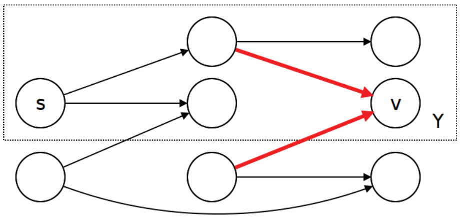
ALGORITHM

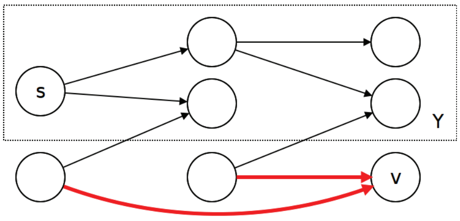
Compute the posterior probability of s being an ancestor of t given the data on a node set N.

A (full) Bayesian averaging approach, based on dynamic programming.

Assumptions: a modular likelihood score, an order-modular structural prior.

Idea: for every node set $X \subseteq N$ and $Y \subseteq X$ compute $g_s(X, Y)$, the contribution of the DAGs on X that have a directed path from s to all $u \in Y$ and to no other node.





Time requirement: $O(3^n n^2)$ for all possible pairs s and t.
Space requirement: $O(3^n)$.

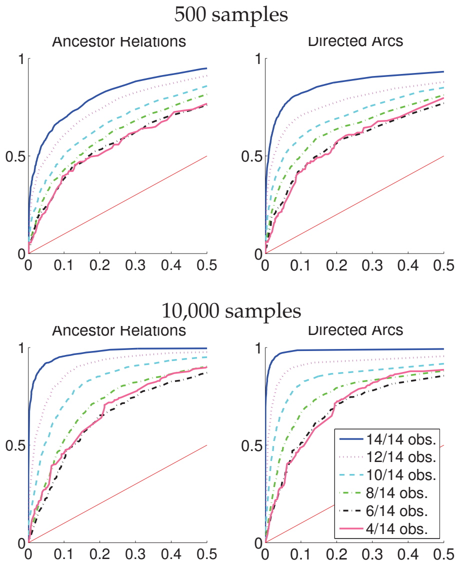
EMPIRICAL RESULTS

Simulation procedure:

- Generate data from a *ground truth*.
- Hide the data on some (unobserved) nodes, form a *shrunken ground truth*.
- Learn ancestor relations from the data on observed nodes.
- Compare the learned ancestor relations to the shrunken ground truth.

Full Bayesian averaging seems to be more powerful than the deducing of ancestor relations from a single MAP DAG or the constraint-based FCI algorithm.

Results with real-life data are in agreement with the simulations.


500 samples

10,000 samples

CONCLUSIONS

Bayesian learning of ancestor relations is computationally feasible (when the number of nodes is moderate).

Ancestor relations can be discovered with reasonable power even in the presence of unobserved variables.

Partial Bayesian averaging, that is, deducing ancestor relations from the arc probabilities seems to work almost as well as full Bayesian averaging.

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

BACKWARD MODEL SELECTION IN FINITE MIXTURE MODELS

Prem Raj Adhikari[1,2] and Jaakko Hollmén[1,2] , {prem.adhikari, jaakko.hollmen}@aalto.fi

[1]Aalto University School of Science, and [2]Helsinki Institute for Information Technology,
Department of Information and Computer Science, PO Box 15400, FI-00076 Aalto, Espoo, Finland

FINITE MIXTURE MODELS

Finite mixture model (FMM) of multivariate Bernoulli distributions are defined as:

$$p(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{J} \pi_j \prod_{i=1}^{d} \theta_{ji}^{x_i} (1-\theta_{ji})^{1-x_i}, \tag{1}$$

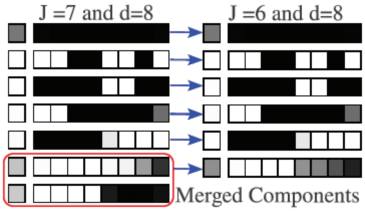
The likelihood function with the model parameters $\{J, \boldsymbol{\pi}, \boldsymbol{\Theta}\}$ is

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} log \left[\sum_{j=1}^{J} \pi_j \prod_{i=1}^{d} \theta_{ji}^{x_{ni}} (1-\theta_{ji})^{1-x_{ni}} \right]. \tag{2}$$

MODEL SELECTION

- If number of components, J, is known a priori EM algorithm can be used to maximize the log-likelihood
- Model selection aims at selecting an appropriate J as it is unknown
- Trade-off between complex models (large J often a reason for Overfitting) and simple models (small J, often a reason for Underfitting)
- Cross-validated likelihood can be used as a model selection criterion
- We choose 10-fold cross validation
- Other criterion such as Penalized likelihood, AIC, BIC, MDL could be used
- Aim is to achieve maximally simple, and compact parsimonious models.

MERGING MIXTURE COMPONENTS



Progressively merge mixture components having minimum KL divergence using Equation 3 and their parameters using Equation 4

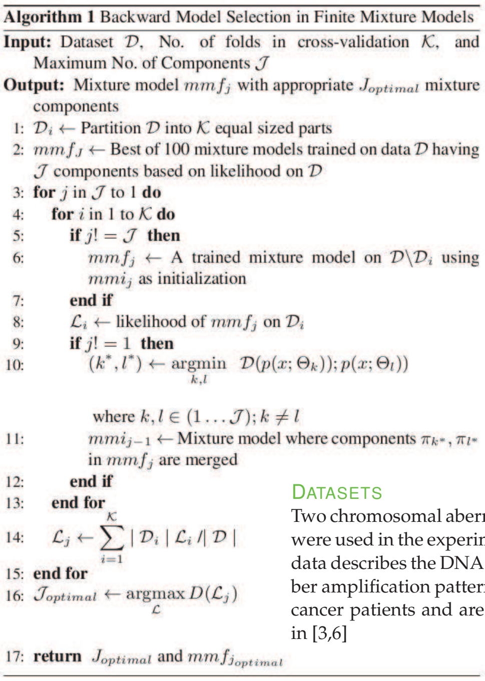
$$\pi_{merged} = \pi_{klmin,1} + \pi_{klmin,2} \tag{3}$$

Here π_{merged} is the merged component and $\pi_{klmin,1}$ and $\pi_{klmin,2}$ are the two candidate components with minimum KL divergence selected to merge.

$$\Theta_{merged} = \frac{\pi_{klmin,1} \times \Theta_{klmin,1} + \pi_{klmin,2} \times \Theta_{klmin,2}}{\pi_{klmin,1} + \pi_{klmin,2}}$$

In Equation 4, Θ_{merged} are the parameter vectors of the component π_{merged} obtained by merging two components in Equation 3. Similarly, $\Theta_{klmin,1}$ and $\Theta_{klmin,2}$ are the parameter vectors of the two components having minimum KL divergence selected for merging.

PROPOSED ALGORITHM



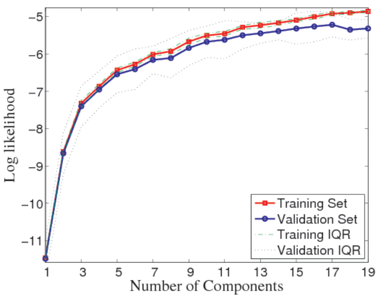
Algorithm 1 Backward Model Selection in Finite Mixture Models

```
Input: Dataset $\mathcal{D}$, No. of folds in cross-validation $\mathcal{K}$, and
    Maximum No. of Components $\mathcal{J}$
Output: Mixture model $mmf_j$ with appropriate $J_{optimal}$ mixture
    components
$\mathcal{D}_i \leftarrow$ Partition $\mathcal{D}$ into $\mathcal{K}$ equal sized parts
$mmf_J \leftarrow$ Best of 100 mixture models trained on data $\mathcal{D}$ having
    $\mathcal{J}$ components based on likelihood on $\mathcal{D}$
for $j$ in $\mathcal{J}$ to 1 do
    for $i$ in 1 to $\mathcal{K}$ do
        if $j! = \mathcal{J}$ then
            $mmf_j \leftarrow$ A trained mixture model on $\mathcal{D} \backslash \mathcal{D}_i$ using
            $mmi_j$ as initialization
        end if
        $\mathcal{L}_i \leftarrow$ likelihood of $mmf_j$ on $\mathcal{D}_i$
        if $j! = 1$ then
            $(k^*, l^*) \leftarrow \underset{k,l}{\text{argmin}}\ \mathcal{D}(p(x;\Theta_k)); p(x;\Theta_l))$
            where $k, l \in (1 \ldots \mathcal{J}); k \neq l$
            $mmi_{j-1} \leftarrow$ Mixture model where components $\pi_{k^*}, \pi_{l^*}$
            in $mmf_j$ are merged
        end if
    end for
    $\mathcal{L}_j \leftarrow \sum_{i=1}^{\mathcal{K}} | \mathcal{D}_i | \mathcal{L}_i /| \mathcal{D} |$
end for
$\mathcal{J}_{optimal} \leftarrow \underset{\mathcal{L}}{\text{argmax}}\ D(\mathcal{L}_j)$
return $J_{optimal}$ and $mmf_{j_{optimal}}$
```

DATASETS

Two chromosomal aberration data were used in the experiments. The data describes the DNA copy number amplification pattern of 4590 cancer patients and are same as in [3,6]

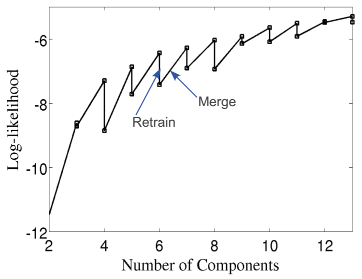
NUMBER OF COMPONENTS(J)



We use 10-fold cross validation over different components

IMPORTANCE OF RETRAINING

Trajectories of Log-likelihood



KULLBACK LEIBLER DIVERGENCE BETWEEN MIXTURE COMPONENTS

In a mixture model, the KL divergence between two mixture components can be derived to

$$KL_{\theta\beta} = \sum_{i=1}^{2^d} \left[\left\{ \prod_{k=1}^{d} \left(\theta_k^{X_{ik}} (1-\theta_k)^{(1-X_{ik})} \right) - \prod_{k=1}^{d} \left(\beta_k^{X_{ik}} (1-\beta_k)^{(1-X_{ik})} \right) \right\} \cdot log \prod_{k=1}^{d} \frac{\theta_k^{X_{ik}} (1-\theta_k)^{(1-X_{ik})}}{\beta_k^{X_{ik}} (1-\beta_k)^{(1-X_{ik})}} \right]$$

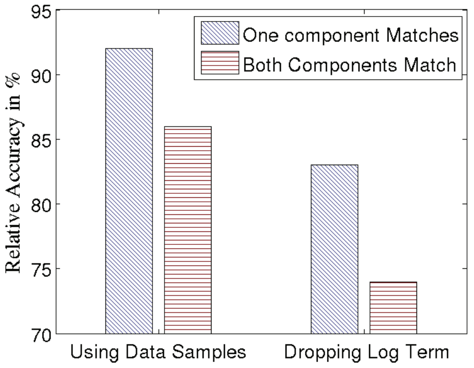
We derive data driven approximation of KL divergence as

$$KL_{\theta\beta} = \sum_{X^* \subset \underline{X}} \left\{ \prod_{k=1}^{d} \left(\theta_k^{X_{ik}} (1-\theta_k)^{(1-X_{ik})} \right) - \prod_{k=1}^{d} \left(\beta_k^{X_{ik}} (1-\beta_k)^{(1-X_{ik})} \right) \right\}$$

APPROXIMATIONS USED

- Dropping the log-term : $log \frac{0}{0} \approx 0$
- Using only unique samples in the data instead of full state-space
- Approximating state-space by unique samples $X^* \subset \underline{X}$ provides data driven approach of approximation of KL divergence

APPROXIMATIONS IN KL DIVERGENCE



REFERENCES

1. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.

2. P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10:63–72, 2000.

3. J. Tikka, J. Hollmén, and S. Myllykangas. Mixture Modeling of DNA copy number amplification patterns in cancer. In Francisco Sandoval, Alberto Prieto, Joan Cabestany, and Manuel Graña, editors, *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, volume 4507 of Lecture Notes in Computer Science, pages 972–979, San Sebastián, Spain, 2007. Springer-Verlag.

4. N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM Algorithm for Mixture Models. *Neural Computation*, 12(9):2109–2128, 2000.

5. S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Maths. Stat.*, 22(1):79-86, 1951.

6. S. Myllykangas, J. Tikka, T. Boȟling, S. Knuutila, and J. Hollmń. Classification of human cancers based on DNA copy number amplification modeling. *BMC Medical Genomics*, 1(15), May 2008.

7. M. Baudis. Genomic imbalances in 5918 malignant epithelial tumors: an explorative meta-analysis of chromosomal CGH data. *BMC Cancer*, 7:226, December 2007.

Aalto University
School of Science

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

ALGODAN Algorithmic Data Analysis

ACKNOWLEDGEMENT

The work is funded by Helsinki Doctoral Programme in Computer Science - Advanced Computing and Intelligent Systems(**Hecse**).

Environmental Proxy Selection Problems in Temperature Reconstruction

Mikko Korpela[1,2,3] and Jaakko Hollmén[1,3]
{mikko.korpela, jaakko.hollmen}@aalto.fi
[1]Aalto University School of Science, Department of Information and Computer Science
[2]University of Helsinki, Department of Computer Science
[3]Helsinki Institute for Information Technology HIIT

Introduction

Direct temperature measurements are only available from the past few hundred years. Therefore, proxy measurements must be used. We study the use of different environmental proxy variables for temperature reconstruction. Differences in both the time coverage of the proxies (Fig. 1) and the temperature signal present in them pose a challenge to the recovery of reliable temperature records (Fig. 2).

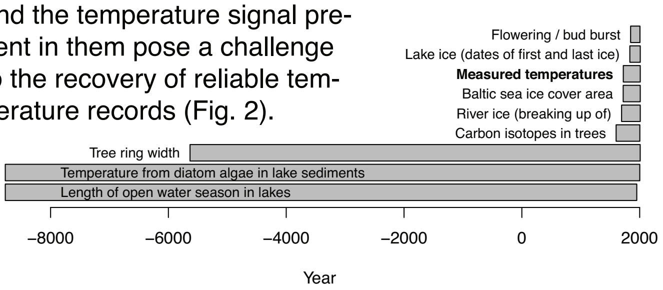


Fig. 1: Rough availability of different proxy measurements

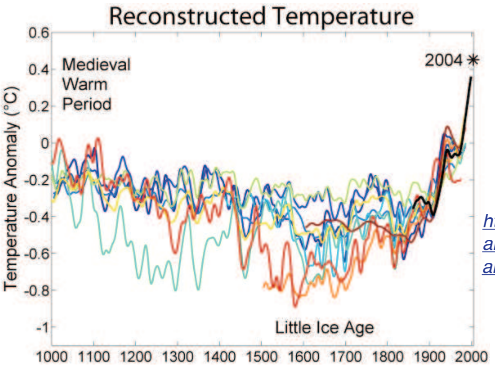


Fig. 2: Different temperature reconstructions. Image created by Robert A. Rohde / Global Warming Art. http://www.globalwarmingart.com/wiki/File:1000_Year_Temperature_Comparison.png

Environmental Proxy Selection Problem

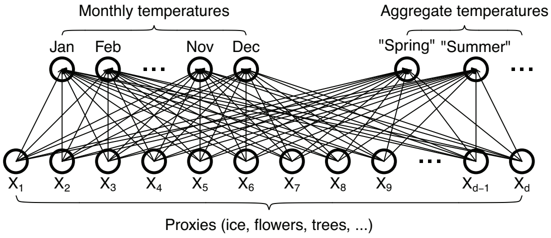


Fig. 3a: Full model (no variable selection)

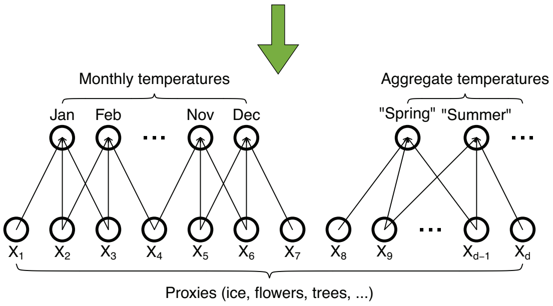


Fig. 3b: Most informative proxies found (variable selection performed)

- Identify the most informative proxy variables for reconstruction of temperature in Finland (Fig. 3)
- Different time of year or different geographic location ⇒ alternative set of good proxies
- Search based solutions, e.g. working on an R version of the backward selection type algorithm SISAL [2].
  - Extend [2] by exploring more states by branching
  - Issues to solve: ill-conditioned problem when number of variables is small compared to number of samples (Fig. 4, Fig. 5), ...

The dplR package for R

The dendrochronology program library in R (dplR) [1] is an add-on package for the R Project for Statistical Computing. These are open source software.

We use the package for preprocessing of tree ring measurements and do active development to make it better suit our needs. Some of our contributions include:

- Improved performance
- Bug fixes, especially corner cases
- Support for additional data formats (e.g. TRiDaS)
- Other new functionality (example below)

Name	Tree	Core
P0101A	101	1
P0101B	101	2
P0102A	102	1
...		
536011	1	1
536012	1	2
536021	2	1
536022	2	2

Problem

In tree ring databases, metadata is scarce. Need to identify which measurements are from the same tree. Manual labeling is cumbersome.

Solution

Derive tree and measurement (=core) IDs from record names (in very uncertain cases, use correlations, too). Function 'autoread.ids' does this automatically. Intelligent discovery of naming schemes, fixes small typos.

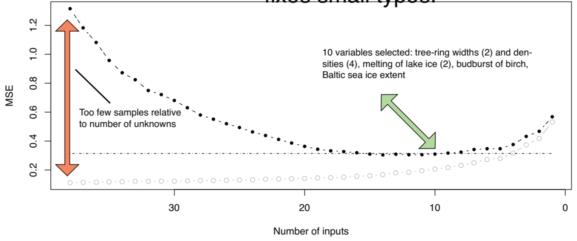


Fig. 4: Progress of backward selection type algorithm (SISAL with branching). Training (smaller values) and validation (larger values) error.

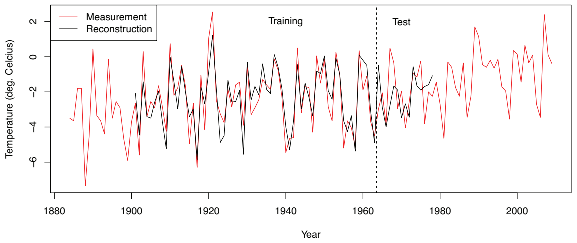


Fig. 5: Temperature reconstruction (mean of April and May temperatures in Jyväskylä) with the model of Fig. 4. Adjusted $R^2 = 0.74$.

References

[1] Andrew G. Bunn. A dendrochronology program library in R (dplR). *Dendrochronologia,* 26(2):115–124, 2008.

[2] Jarkko Tikka, Jaakko Hollmén. Sequential Input Selection Algorithm for Long-term Prediction of Time Series. *Neurocomputing,* 71(13–15):2604–2615, 2008.

Aalto University
School of Science

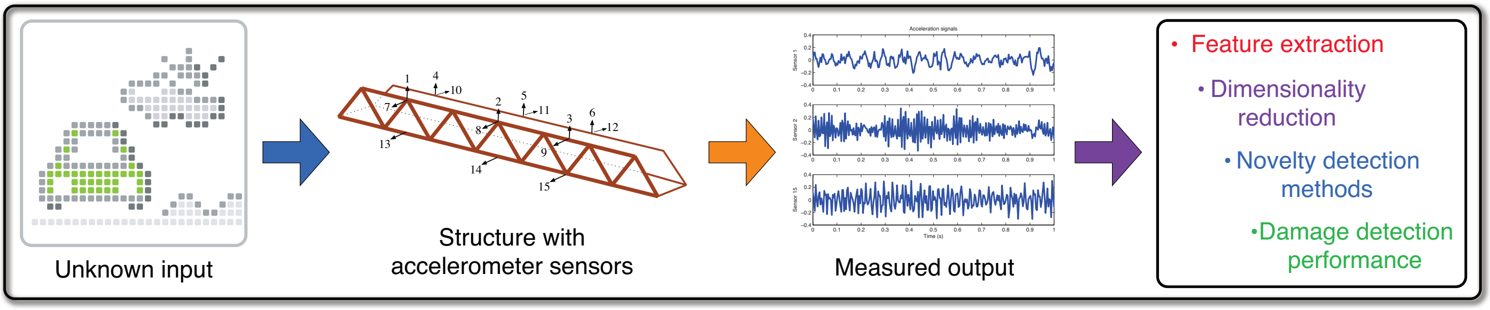
Damage detection methods for Structural Health Monitoring with Wireless Sensor Networks

Janne Toivola

janne.toivola@aalto.fi

Department of Information and Computer Science, Aalto University, Finland

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY



Unknown input

Structure with accelerometer sensors

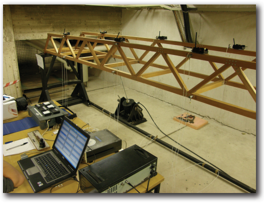
Measured output

- Feature extraction
  - Dimensionality reduction
    - Novelty detection methods
      - Damage detection performance

Vibration-based SHM

- Structural Health Monitoring: assessing the condition of physical structures
- Damages are assumed to change the structure as a medium for vibrations caused by the environment.
- Wired sensors: expensive to maintain for large structures
- Wireless sensors: limited energy and bandwidth

Wooden model bridge



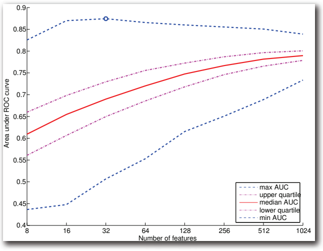
- Controlled test environment
- Input: electronic shaker
- Output: time series data from wireless and wired accelerometers
- Hardware and software developed in the multidisciplinary ISMO project

Damage detection based on machine learning methods

- Models based on acquired data: avoid complex physics-based models, geometry etc.
- Parsimonius detection algorithms required for online computation
- Dependencies between data from separate sensors are important for detecting damages in a structure

Feature extraction

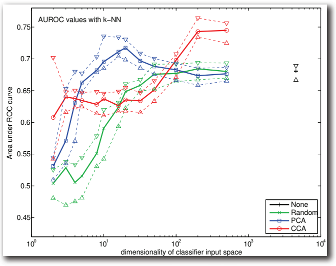
- Online frequency domain features with the Goertzel algorithm
  - Running on a WSN node: [Bocca: ICCPS 2011]
- Transmissibility: propagation of vibrations between two sensors
  - Large, but redundant feature space:



[Toivola: IDA 2009]

Dimensionality reduction

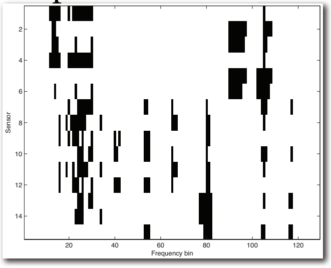
- Combinations of projection and novelty detection methods assessed for accuracy:



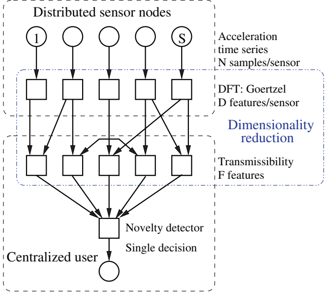
[Toivola: IDA 2010]

- Three-way analysis over time, sensor pairs, and vibration frequency: [Prada: IEEE MLSP 2010]

- Collaborative filtering method for using SHM-specific locally computed ratings for selecting a global set of sparse features:



[Toivola: ICDMW 2011]



Novelty detection methods

- Nearest neighbor vs Parzen, Gaussian, and Mixture of Gaussians density models
- Static: independent detections across time

Performance

- Accuracy assessed in terms of ROC AUC, energy in terms of feature vector length
- Alternative: *change detection* framework
  - Better criteria for accuracy and energy efficiency..?

Acknowledgements

- Jaakko Hollmén: supervisor
- Jyrki Kullaa: data & expertise
- Miguel A. Prada: projections and three-way analysis
- Maurizio Bocca: embedded WSN implementations
- Hecse: funding conf. trips
- MIDE / Aalto University: funding ISMO project
- Algodan Centre of Excellence

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

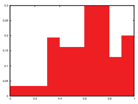
Density and Entropy Estimation with NML Histograms

Panu Luosto, Ciprian Doru Giurcăneanu and Petri Kontkanen

We compare empirically four histogram methods for density and entropy estimation. They include the *normalized maximum likelihood* (NML) histogram by Kontkanen and Myllymäki, and its novel variant that is based on NML as well. As an extension to irregular histograms, we also test the new MDL based *clustgram*.

Basic vocabulary

- **Irregular histogram.** A histogram in which the widths of the bins are not necessarily equal.



- **How many bins?** The model class selection problem is here to choose the most appropriate number of bins.
- **Risk minimization.** A statistical approach, in which the goal is to minimize e.g. the KL or squared Hellinger distance from an assumed unknown true distribution to the estimated distribution.
- **Minimum description length (MDL) principle.** An information-theoretic criterion that does not require that a true distribution should exist. The best model class is the one that allows the most effective encoding of the data.
- **Normalized maximum likelihood (NML).** Maximum likelihoods turned into a distribution through normalization:

$$p_{\mathrm{NML}}(\mathbf{x}) = \frac{p(\mathbf{x};\, \hat{\theta}(\mathbf{x}))}{\sum_{\mathbf{y}\in X^n} p(\mathbf{y};\, \hat{\theta}(\mathbf{y}))},$$

where $\hat{\theta}(x^n)$ refers to the maximum likelihood parameters of $\mathbf{x} \in X^n$. If a NML distribution exists, it minimizes the worst-case excess code length compared to the optimal code length in hindsight (only an oracle can guess the ML parameters before seeing the data).

Methods

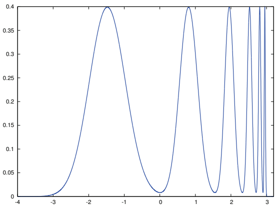
Methods that choose the bin borders from a regular grid:

- **NML-1**: the histogram of Kontkanen and Myllymäki (2007), optimizing the choice of k bins.
- **NML-2**: a new version of the former, optimizing the choice of k *non-empty* bins. We also optimized the choice of the grid.

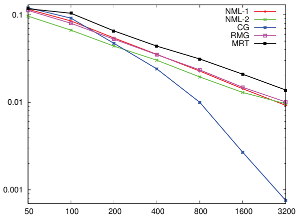
Methods that choose the bin borders from the set of data points:

- **RMG**: a method of Rozenholc, Mildenberger and Gather (2010). Based on Massart's results about risk bounds (2007) and on empirical considerations.
- **MRT**: a method of Menez, Rendas and Thierry (2008). The penalty is BIC plus a combinatorial term.
- **Clustgram**: an MDL-based extension to irregular histograms by Luosto and Kontkanen (2011) with many types of bins: uniform, normal, half-normal, exponential and Laplace.

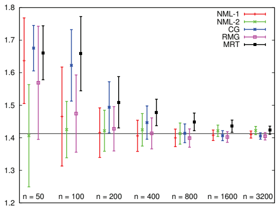
Example 1



Mixture with 6 normal components.

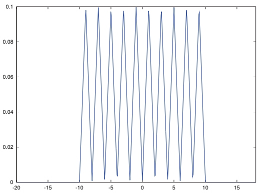


Hellinger distances to the source distribution.

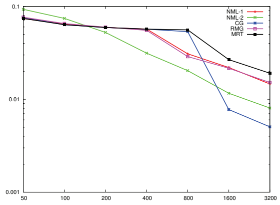


Estimated entropies (mean and standard deviation). The true entropy is indicated with a horizontal line.

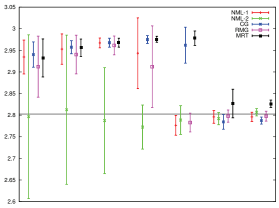
Example 2



Mixture with 10 triangular components.



Hellinger distances to the source distribution.



Estimated entropies (mean and standard deviation). The true entropy is indicated with a horizontal line.

Conclusions

The performance of NML-1 was in terms of the squared Hellinger distance similar to RMG, which has been specially designed to minimize the statistical risk. The novel NML-2 seemed to work especially well with ragged multimodal distributions.

Aalto University

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Metabolite identification and molecular fingerprint prediction via machine learning

Markus Heinonen[1,3*], Huibin Shen[1], Nicola Zamboni[4], Juho Rousu[2,3]
[1] Department of Computer Science, University of Helsinki, Finland
[2] Department of Inofmration and Computer Science, Aalto University, Finland
[3] Helsinki institute for Information Technology, Finland
[4] Institute of Molecular Systems Biology, ETH Zurich, Switzerland

ABSTRACT

Identification of metabolites from tandem mass spectrometry measurements is a prerequisite step for metabolic modeling and network analysis. Currently this task requires matching of measured mass spectra against annotated databases of reference spectra, and extensive manual work. We propose a machine learning framework, which identifies the metabolite structures based on the mass spectral signals. Our approach is twofold (see Fig 1). First, (1) we decompose the problem into binary subproblems each predicting an individual structural property of the unknown structure. Then, (2) the complete structure is inferred from the predicted fingerprints by searching candidate molecules from databases matching these fingerprints. The method's performance is shown with experiments using several real-life mass spectral datasets.

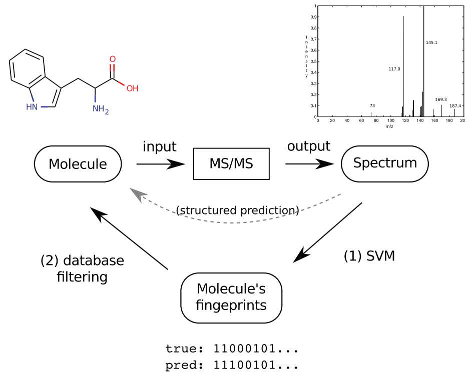


Figure 1: Metabolite identification scheme. Instead of directly predicting the molecule from measured spectrum (structured prediction), we opt to predict an intermediate target of a fingerprint vector, which is subsequently used to pinpoint the molecule from a molecular database.

FINGERPRINTS

We predict as intermediate targets molecular fingerprints, which are binary descriptors of a molecule. We use 528 structural fingerprints, e.g. "does the molecule contain an amino-group", "does the molecule contain a double bond", etc.

(1) SVM & KERNELS

Let an input mass spectrum $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{X}$ be a collection of k peaks $\mathbf{x}_i \in \mathbb{R}^2$. A peak tuple $\mathbf{x} = (mass, int)^T$ represents the mass-to-charge ratio and the intensity of measured peak.

We use SVM to predict m binary fingerprints $(y_i)_{i=1}^m = \mathbf{y}$ as independent classification tasks $f_i : \mathcal{X} \to \{0,1\}$. In SVM a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defines a feature mapping $\phi(\chi)$ such that $K(\chi, \chi') = \langle \phi(\chi), \phi(\chi') \rangle$.

We experiment with a simple discrete kernel and also with a high resolution continuous probability product kernel. First, we represent spectra χ and χ' with probabilistic models p and p', respectively. Then, we define the kernel similarity of spectra χ and χ' as similarity between the corresponding distributions p and p' as

$$K(\chi, \chi') \equiv K(p, p') = \int_{\mathbb{R}^2} p(\mathbf{x}) p'(\mathbf{x}) d\mathbf{x}$$

A natural probabilistic distribution over the set of peaks is a gaussian model $p(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^k p_i(\mathbf{x})$, where each peak contributes density according to a gaussian $p_i \sim \mathcal{N}(\mathbf{x}_i, \Sigma)$ centered at the peak $(mass, int)$ (see Fig 3).

We use three classes of mass spectral features, and their combinations:

- *peaks* $\{\mathbf{x}_i\}$
- *neutral losses* $\{(prec - mass_i, int_i)^T\}$ measure the mass distance from the precursor peak
- *peak differences* $\{\mathbf{x}_i - \mathbf{x}_j : \forall i < j\}$ measure the mass distance between any two peaks

(2) DATABASE FILTERING

The SVM learns a mapping from the spectral features to individual structural characteristics $\mathbf{y}$ of the measured molecule. We employ the fingerprints as filters on molecular repositories, such as PubChem, which contain millions of molecules. The candidate molecules are suggested according to the Poisson-Binomial probability of the fingerprint prediction, given the crossvalidation accuracies $\mathbf{p} = (p_i)_{i=1}^m$

$$p(\hat{\mathbf{y}} | \mathbf{p}, \mathbf{y}) = \prod_{i=1}^m p_i^{1-|\hat{y}_i - y_i|} (1 - p_i)^{|\hat{y}_i - y_i|}.$$

EXPERIMENTS

We conducted experiments on predicting 528 fingerprints of three mass spectral datasets, containing 514, 403 and 293 molecule-spectrum pairs, respectively. We trained an SVM using the probability product kernel for each fingerprint individually using 5-fold crossvalidation. The average fingeprint prediction accuracies for the three datasets were 91.1%, 91.1% and 99.5% with baselines of 87.3%, 78.7% and 88.3%, respectively.

Figure 4 indicates the individual fingerprint prediction accuracies using the two kernels on a high resolution mass spectral dataset. Figure 5 indicates the ROC curves indicating the proportions of data achieving certain identification ranks.

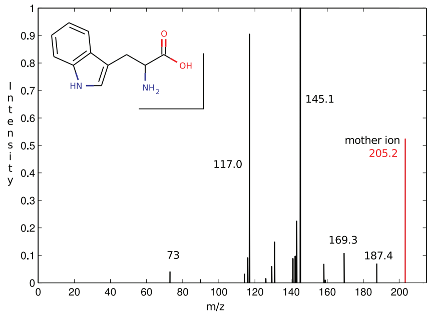


Figure 2: MS/MS spectrum of Tryptophan (mass 204.23). Each peak represents the mass of a fragment of Tryptophan. The red peak indicates the non-fragmented mother ion.

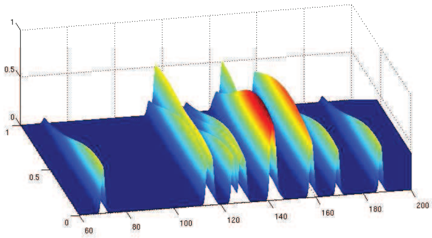


Figure 3: The 2D gaussian mixture density of mass spectrum of Fig 2.

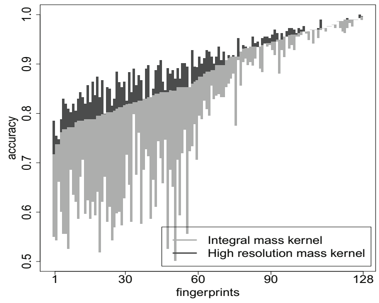


Figure 4: Prediction accuracies of individual fingerprints using two different kernels.

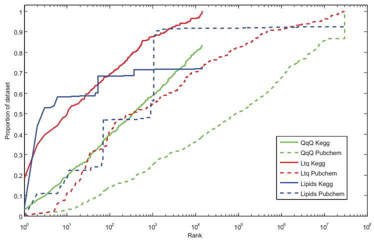


Figure 5: The rank of the correct metabolite in our prediction. The colors indicate three different datasets, while the linetype indicates querying from either PubChem database (largest repository of molecules) or from KEGG (a small database of metabolites).

Heinonen, M., Shen, H., Zamboni, N., Rousu, J. *Metabolite identification and molecular fingerprint prediction via machine learning.* ECCB'12, submitted.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Efficient Path Kernels for Reaction Function Prediction

Markus Heinonen*, Niko Välimäki, Veli Mäkinen, Juho Rousu
{markus.heinonen, niko.valimaki, veli.makinen, juho.rousu}@cs.helsinki.fi
Department of Computer Science, University of Helsinki, Finland

ABSTRACT

We propose the first efficient path-based graph kernel for classification of reaction graphs. The path kernel utilizes efficient compressed path index data structure. In our experiments we outperform state-of-the-art graph kernels in prediction of the EC code of organic reactions.

KERNELS

We consider labeled undirected graphs $G = (V, E, L)$, where a labeling function L applies to both nodes $v \in V$ and edges $(v, u) \in E$. A *walk* w is a sequence of adjacent vertices, possibly infinite. A *path* p is a finite walk with no repeats. A graph kernel $K : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ is a positive semi-definite similarity function over pairs of graphs, which implicitly defines some feature mapping $\phi(G)$ such, that $K(G, G') = \langle \phi(G), \phi(G') \rangle$. Sequence based graph kernels are

$$K_{walks}(G, G') = \sum_{w \in \mathcal{W}} \lambda^{|w|} \phi(G)_w \cdot \phi(G')_w$$

$$K_{paths}(G, G') = \sum_{p \in \mathcal{P}} \lambda^{|p|} \phi(G)_p \cdot \phi(G')_p$$

$$K_{sp}(G, G') = \sum_{p \in \mathcal{SP}} \lambda^{|p|} \phi(G)_p \cdot \phi(G')_p,$$

where length decay $\lambda < 1$, $\phi(G)_w$ is count of walk w in G and $\mathcal{W}$, $\mathcal{P}$ and $\mathcal{SP}$ are the universes of walks, paths and shortest paths.

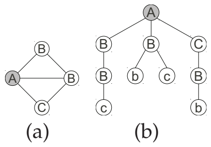
COMPUTING KERNELS VIA COMPRESSED PATH INDEX

1. Enumeration of paths. We traverse the graphs using depth-first search and enumerate all paths up to length k.

2. Path index construction. We store the set of paths as a path-sorted XBWT.

3. Computing path frequencies.

4. Computing the kernel. The kernel matrix K is computed as a dot product between path frequency vectors $\phi(G)$ and $\phi(G')$.



(a) (b)

	S_{last}	S_α	S_π
1	0	A	*empty*
2	0	B	A
3	0	B	A
4	1	C	A
5	1	B	BA
6	0	b	BA
7	1	c	BA
8	1	c	BBA
9	1	b	BCA
10	1	B	CA

(c)

Figure 1: (a) Example graph. (b) Paths originating from node A. (c) An XBWT representation of the tree in (b). The rows are lexicographically sorted.

REACTION GRAPHS

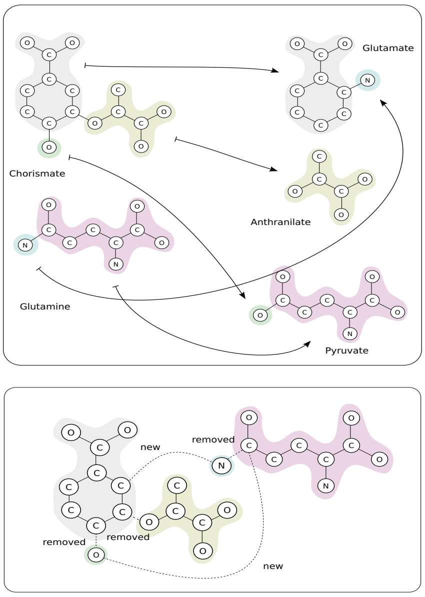


Figure 2: Reaction `R00986 Chorismate pyruvate-lyase` with EC code 4.1.3.27. The reaction and its atom correspondences highlighted with color (top). The reaction graph produced by taking the union of edges from both sides (bottom).

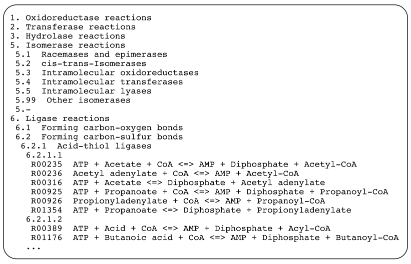
EXPERIMENTS

We evaluate the performance of the various graph kernels on a task, where the EC number of a reaction is predicted. The EC number is a hierarchical code defining the semantic function of the enzymatic reaction. Our dataset is 17430 reactions from KEGG database, with graphs $G_1, \ldots, G_{17430}$. The EC hierarchy of each reaction is encoded as 270 binary targets variables $Y_1, \ldots, Y_{270}$. Our classification task is to predict the three level EC code as a binary multiclassification problem. A result was deemed correct if the correct root-to-leaf branch is predicted.

We ran the experiments with MMCRF hierarchical multilabel classification algorithm. All kernels use $\lambda = 0.90$ and are quadratic kernels as they acchieved consistently best results. A five-fold cross-validation procedure was used.

We experimented with upper bounds of 15 and 50 on the path lengths. We also experimented with *core* paths, paths that go through modified edges only. These paths are likely to contain most relevant information regarding the reaction. Finally, we experimented with indicator features, where all features are binary irrespective of the path frequencies.

RESULTS



```
1. Oxidoreductase reactions
2. Transferase reactions
3. Hydrolase reactions
5. Isomerase reactions
 5.1  Racemases and epimerases
 5.2  cis-trans-Isomerases
 5.3  Intramolecular oxidoreductases
 5.4  Intramolecular transferases
 5.5  Intramolecular lyases
 5.99  Other isomerases
 5.-
6. Ligase reactions
 6.1  Forming carbon-oxygen bonds
 6.2  Forming carbon-sulfur bonds
  6.2.1  Acid-thiol ligases
   6.2.1.1
    R00235  ATP + Acetate + CoA <=> AMP + Diphosphate + Acetyl-CoA
    R00236  Acetyl adenylate + CoA <=> AMP + Acetyl-CoA
    R00316  ATP + Acetate <=> Diphosphate + Acetyl adenylate
    R00925  ATP + Propanoate + CoA <=> AMP + Diphosphate + Propanoyl-CoA
    R00926  Propionyladenylate + CoA <=> AMP + Propanoyl-CoA
    R01354  ATP + Propanoate <=> Diphosphate + Propionyladenylate
   6.2.1.2
    R00389  ATP + Acid + CoA <=> AMP + Diphosphate + Acyl-CoA
    R01176  ATP + Butanoic acid + CoA <=> AMP + Diphosphate + Butanoyl-CoA
   ...
```

Figure 3: EC hierarchy consists of 6 main classes, 63 second level and 201 third level categories, a total of 270.

# of reaction graphs	17,430
# of trees	746,438
# of tree nodes	279 mil.
# of tree leaves	91 mil.
max. tree depth	50
Index construction time	1.1 hours
Index construction space	4.4 GB
Final index size	1.1 GB
# of unique paths	21 mil.
Index frequency computation	176 s
Kernel computation (path length 50)	12 min
MMCRF run (average, 5-fold cv)	10 hours

Table 1: Characteristics of the test data and performance results.

Kernel	k	Tr. error (%)	Ts. error (%)
Walk	15	52.9	61.1
RGK	inf	27.8	35.0
Shortest paths		21.5	36.4
Core paths	50	14.9	28.9
Core paths, ind	50	14.5	27.8
All paths	50	19.6	34.2
All paths, ind	50	9.1	25.6
Core paths	15	15.0	28.3
Core paths, ind	15	14.7	27.3
All paths	15	20.0	33.7
All paths, ind	15	9.2	**24.3**

Table 2: Prediction of full EC class. Core path kernel only includes paths with “+1” or “-1” edges, while indicator kernels contain only binary values.

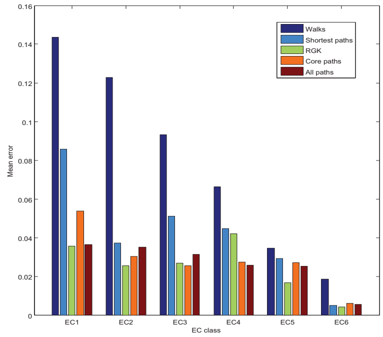


Figure 4: Prediction errors for the six main EC classes.

Heinonen, M., Välimäki, N., Mäkinen, V., Rousu, J. *Efficient Path Kernels for Reaction Function Prediction*. BIOINFORMATICS'12, Vilamoura, Portugal.

Aalto University
School of Science

Protein Interaction Prediction in Yeast based on Sequence Features

Jana Kludas, Juho Rousu
{jana.kludas, juho.rousu}@aalto.fi
Helsinki Institute for Information Technology, Aalto University, Finland

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

INTRODUCTION

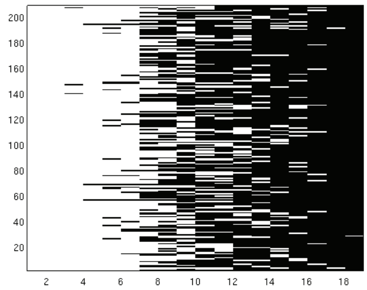
- **protein interactions**: important for system-level understanding of biological processes
- **BIOLEDGE** project: BIO knowLEDGe Extractor and Modeler for Protein Production, focus on **secretion proteins**
- target species: **Saccharomyces cerevisiae**, Pichia pastoris, Trichoderma reesei

RESEARCH GOALS

- in silico prediction of protein interactions based on sequence features
- investigation of biological network reconstruction tools

SEQUENCE FEATURES

Figure 1: Feature coverage for secretion proteins in Saccharomyces cerevisiae with reliable interactions.



- Sparse, High-Dimensional, Few Instances -

ID	NAME	#
1	Blast score	113.798
2	Protein clusters	559
3	GTG	108.810
4	PFAM	860
5	Panther	898
6	Super family	431
7	Gene 3D	360
8	PROSITE Profile	243
9	SMART	231
10	PROSITE Pattern	299
11	PIR FAM	74
12	TIGRFAM	119
13	FINGERPRINT	62
14	PRODOM	20
15	HAMAP	6
		226.770

FEATURE SELECTION

Feature selection has to be performed because modeling with the full fused feature set gives classification accuracies close to random.

ID	NAME	ACCURACY	# FS TOP 1000
1	Blast score	71.3(±3.9)	28.3(±20.1)
2	Protein clusters	71.6(±2.5)	3.7(±1.9)
3	GTG	**74.1**(±**4.3**)	**901.9**(±**39.1**)
4	PFAM	**75.7**(±**5.2**)	10.4(±4.7)
5	Panther	**74.3**(±**3.7**)	7.0(±3.6)
6	Super family	73.7(±5.4)	9.6(±3.2)
7	Gene 3D	73.6(±4.3)	20.1(±7.5)
8	PROSITE Profile	68.6(±2.9)	5.6(±3.7)
9	SMART	70.4(±3.3)	6.1(±4.1)
10	PROSITE Pattern	71.8(±4.3)	4.1(±2.0)
11	PIR FAM	66.9(±0.5)	0.5(±0.8)
12	TIGRFAM	68.5(±1.5)	0.6(±0.7)
13	FINGERPRINT	66.8(±0.4)	1.9(±1.8)
14	PRODOM	66.7(±0.0)	0.0(±0)
15	HAMAP	66.7(±0.0)	0.0(±0)

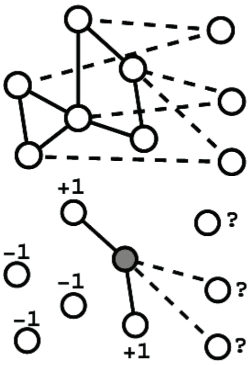
Classification accuracy when the individual feature sets are used for modeling and number of variables included in the top 1000 by feature selection (# FS TOP 1000) when modeling over the fused feature set.

PROTEIN-PROTEIN-INTERACTION (PPI) PREDICTION

Given a set of proteins $V = (v_1, ..., v_n)$,
a set of feature vectors $\Phi(v_1), ..., \Phi(v_n) \in \Re_p$,
a set of known interactions $S = ((e_1, y_1), ..., (e_m, y_m))$
as pairs of vertices: $e_i \in V \times V$ with $y_i = [1; -1]$.

INFERENCE WITH LOCAL MODELS

1. choose a seed vertex $v_{seed} \in V$
2. create local training set
3. feature selection with Mutual Information measure over all feature-label pairs
4. train SVM on the local training set
5. predict label of any vertex that has no label
6. repeat step 1.-6. for each vertex $v_{seed} \in V$
7. combine the predicted edges



LABELS: PROTEIN INTERACTIONS IN STRING

STRING is a data base of known and predicted protein interactions (`string-db.org`). Links are given as probability scores $[0, .., 1000]$ for genetic neighborhood, fusion, co-occurrence, co-expression, experiments, databases, text mining as well as a combined score (cs).

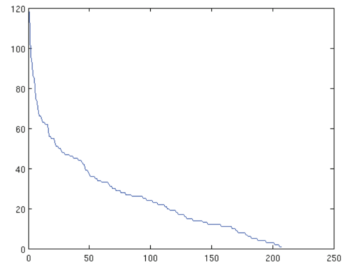
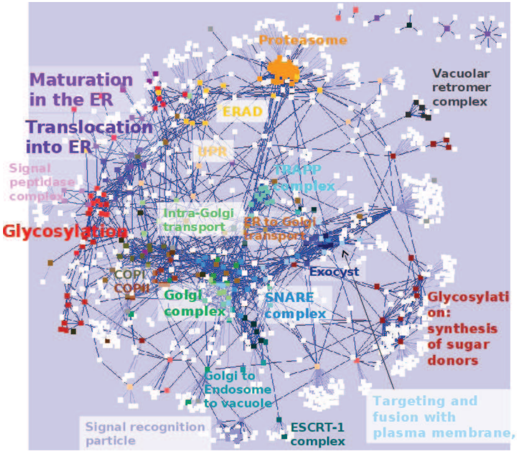


Figure 2: Number of reliable interactions (cs>500) for secretion proteins in Saccharomyces cerevisiae.

Figure 3: PPI network of secretory pathway proteins in Saccharomyces cerevisiae (by M. Oja@VTT).



CLASSIFICATION EXPERIMENT

SVM with RBF kernel (LibSVM package v.3.12)

TRAINING/EVALUATION DATASET

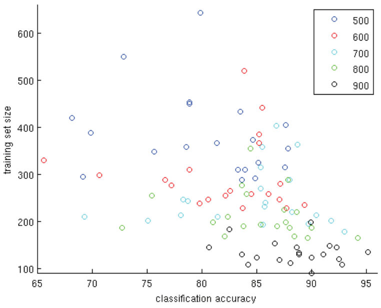
(with known ground truth)
N_{pos} examples: reliable interactions $cs > [500, .., 900]$
N_{neg} examples: random selection of probably non interacting proteins, $N_{neg} = 2N_{pos}$

CROSS VALIDATION

10 folds of train 80%/ test 20%
(I) for model selection (σ, C, feature selection)
(II) for model evaluation

RESULTS

+ inference with local models gives accurate results when trained on reliable interactions
- choice of seed vertices is limited to proteins with enough known interactions



CS	ACCURACY	# FS
> 500	80.3(±6.2)	29.300(±5.900)
> 600	82.0(±5.9)	6.300(±10.700)
> 700	84.5(±6.0)	4.000(±3.000)
> 800	85.2(±4.9)	5.000(±10.000)
> 900	**88.5**(±**3.8**)	1.400(±1.000)

CONCLUSIONS AND FUTURE WORK

- local modeling has no good scalability, training a model for each seed is cumbersome
- instead: inference on global models
- visualize interactions predicted on hold out dataset with Cytoscape and evaluate their biological relevance
- improve feature selection
- more experiments i.e. different STRING scores

BIOLEDGE

ACKNOWLEDGEMENTS

The work was financially supported by the BIOLEDGE project (FP7-KBBE-289126), the Helsinki Doctoral Programme in Computer Science (Hecse), Academy of Finland grant 118653 (ALGODAN), and in part by the IST Programme of the European Community under the PASCAL2 Network of Excellence, ICT-2007-216886.

Random Graph Ensembles in Multi-Task Classification

Aalto University
School of Science

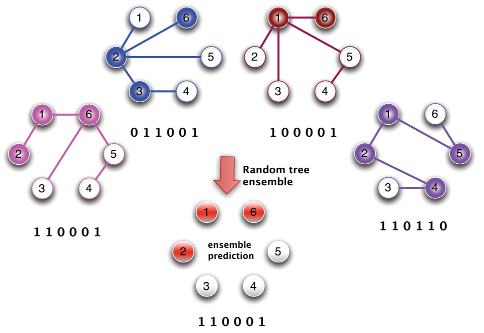
Hongyu Su, Juho Rousu
{hongyu.su, juho.rousu}@aalto.fi
Helsinki Institute for Information Technology, Aalto University, Finland

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

ABSTRACT

We present an ensemble of multi-task classifiers for multilabel classification. As the base classifiers of ensemble, we use Maximum Margin Conditional Random Field (MMCRF) Model. Source diversity of base classifiers arises from the different random output structures, a different approach from boosting or bagging. Experimental result shows that ensembles of random networks outperforms other approaches.

RANDOM GRAPH ENSEMBLE



Input: Training sample $S = \{(x_i, \mathbf{y}_i)\}_{i=1}^m$, ensemble size T, n the number of nodes in the output graph,
Output: Multi-task learner ensemble $\left(f^{(1)}, \dots, f^{(T)}\right)$

```
1: $t = 0$
2: **while** $t < T$ **do**
3:     $t = t + 1$
4:     $G_t = randomGraph(n)$
5:     $f_t = learnBaseClassifier(\{x_i\}_{i=1}^m, (\mathbf{y}_i)_{i=1}^m, G_t)$
6: **end while**
7: $F = (f_1, \dots, f_T)$
```

DATASETS

Table 1: Multilabel datasets from biological and text classification fields used in our empirical studies. Statistics include multilabel density (D), label balance (B) and label correlation (Co).

DATASET	INSTANCES	LABELS	D	B	CO
GENEBASE	662	27	1.25	0.05	0.07
CANCER	4547	60	11.05	0.18	0.73
FINGERPRINT	490	286	49.1	0.17	0.08
ENRON-F	1694	53	3.42	0.06	0.03
SLASHDOT-F	3749	22	1.18	0.05	0.03
LLOG-F	1460	75	1.37	0.02	0.02
WIPO	1710	188	4	0.02	0.01
REUTERS	7500	34	1.48	0.04	0.05
BIBTEX	2515	159	2.43	0.02	0.02
BOOKMARKS	2000	208	2.06	0.01	0.02

CONCLUSION

We have studied the potential of structured output learning on random graphs as the basis of constructing accurate multilabel classification models. Our investigations indicate that models thus created have favorable predictive performance on a heterogeneous collection of multilabel datasets. The results of this paper indicate that structured output prediction methods can be successfully applied to problems where no a priori known output structure exists.

MAX-MARGIN CONDITIONAL RANDOM FIELD (MMCRF)

We consider data from a domain $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a set of objects and $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_k$ is a Cartesian product over the set $\mathcal{Y}_j \in \{+1, -1\}$. A training data set is given as $\{(x_i, \mathbf{y}_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$. A pair $(x_i, \mathbf{y})$ where x_i is a training object and $\mathbf{y}$ is an arbitrary multilabel is called a *pseudo-example*.

JOINT FEATURE MAP

The MMCRF takes a *joint feature map*

$$\phi(x, \mathbf{y}) = (\phi_e(x, \mathbf{y}))_{e \in \mathcal{E}} = \varphi(x) \otimes \psi(\mathbf{y}),$$

where $\otimes$ is the tensor product over input feature map $\varphi(x)$ and output feature map $\psi(\mathbf{y})$.

MODEL FAMILY

As model family, we use exponential family

$$p(y|x) = \frac{1}{Z(x_i, \mathbf{w})} \prod_{e \in \mathcal{E}} \exp(\mathbf{w}^T \phi_e(x, \mathbf{y}_e))$$

defined on edges $e \in \mathcal{E}$ of a Markov network $\mathcal{G}$.

MAX MARGIN LEARNING

Margin-based learning takes the form

$$\mathbf{w} = \underset{\mathbf{w}}{\textbf{argmin}} \left(\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i \right)$$

$$\textbf{s.t. } \mathbf{w}^T \Delta\phi(x_i, \mathbf{y}) \geq \ell_\Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \forall i, \mathbf{y},$$

where $\Delta\phi(x_i, \mathbf{y}) = \phi(x_i, \mathbf{y}_i) - \phi(x_i, \mathbf{y})$, and $\ell_\Delta(\mathbf{y}_i, \mathbf{y})$ encodes the loss of the pseudo-example, as shown in left part of Figure 1.

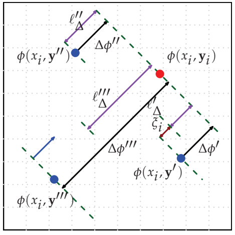


Figure 1: Maximum margin optimization.

Intuitively, it maximizes the margins between the real example and the pseudo-examples. The margins are scaled according to loss function $\ell_\Delta(\mathbf{y}_i, \mathbf{y})$.

MAKING PREDICTIONS

Once we get the edge-labeling specific feature weight $\mathbf{w}$, we can make prediction by maximizing the scoring function

$$\hat{\mathbf{y}}(x) = \underset{\mathbf{y} \in \mathcal{Y}}{\textbf{argmax}}\, \mathbf{w}^T \phi(x, \mathbf{y}).$$

EXPERIMENTAL RESULTS

Table 2: Multilabel loss (top) and Hamming loss (bottom) with standard deviation of the different classification methods.

DATASET	MULTILABEL LOSS			
	ENSEMBLE	SINGLE	SVM	MLKNN
GENEBASE	**1.8 ± 1**	**1.8 ± 1**	2.1 ± 1.1	8.6 ± 2.1
CANCER	61.5 ± 2	64.5 ± 1.2	66.1 ± 1.5	**55.9 ± 1.6**
FINGERPRINT	**95.7 ± 1.8**	95.9 ± 1.6	96.7 ± 0.9	100 ± 0
ENRON-F	**86 ± 0.8**	86.2 ± 0.9	87.2 ± 1.3	90.1 ± 1.4
SLASHDOT-F	75.8 ± 1.8	76.6 ± 1.4	**72.9 ± 1.8**	78.1 ± 1.1
LLOG-F	**78.7 ± 1.3**	78.8 ± 1.4	79.7 ± 1.2	81.5 ± 1.5
WIPO	**72 ± 2.4**	72.2 ± 2.4	74.4 ± 1.8	80.3 ± 2.7
REUTERS	**31.8 ± 0.7**	32.1 ± 0.8	32 ± 1.2	35.1 ± 3.2
BIBTEX	85.6 ± 2	86.2 ± 2.2	**84.6 ± 1.3**	86 ± 1.2
BOOKMARKS	**83.2 ± 2**	83.3 ± 1.7	84 ± 2.3	84.7 ± 2.3
average	**67.2 ± 1.6**	67.8 ± 1.5	68 ± 1.4	70 ± 1.7

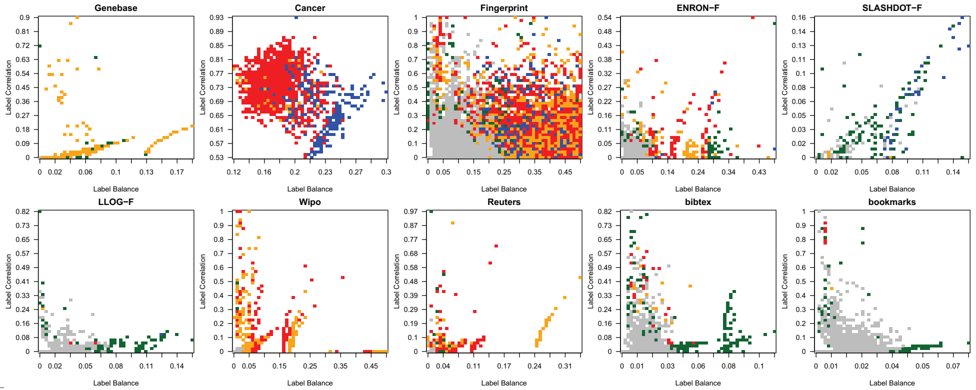
DATASET	HAMMING LOSS			
	ENSEMBLE	SINGLE	SVM	MLKNN
GENEBASE	**0.1 ± 0**	**0.1 ± 0**	**0.1 ± 0.1**	0.4 ± 0.1
CANCER	**13.6 ± 0.4**	13.8 ± 0.3	13.8 ± 0.5	15.7 ± 0.3
FINGERPRINT	**10.2 ± 0.7**	**10.2 ± 0.6**	**10.2 ± 0.3**	11 ± 0.7
ENRON-F	4.8 ± 0.1	4.9 ± 0.1	**4.6 ± 0.1**	4.9 ± 0.1
SLASHDOT-F	6.5 ± 0.2	6.7 ± 0.2	**4.4 ± 0.1**	4.7 ± 0
LLOG-F	1.9 ± 0.1	1.9 ± 0.1	**1.6 ± 0**	**1.6 ± 0**
WIPO	**0.9 ± 0**	**0.9 ± 0**	**0.9 ± 0**	1 ± 0
REUTERS	**1.8 ± 0**	**1.8 ± 0**	**1.8 ± 0**	2.2 ± 0
BIBTEX	1.6 ± 0.1	1.6 ± 0.1	**1.3 ± 0**	**1.3 ± 0**
BOOKMARKS	1.2 ± 0	1.2 ± 0.1	**0.9 ± 0**	**0.9 ± 0**
average	4.28 ± 0.2	4.3 ± 0.2	**4 ± 0.1**	4.4 ± 0.1

Table 3: F_1 score (top) and balanced accuracy (bottom) with standard deviation of the different classification methods.

DATASET	F_1			
	ENSEMBLE	SINGLE	SVM	MLKNN
GENEBASE	**99.2 ± 0.3**	**99.2 ± 0.3**	98.9 ± 0.6	95.1 ± 1.7
CANCER	**59.7 ± 2.2**	59.4 ± 1.9	54.8 ± 2.8	41 ± 3.2
FINGERPRINT	**67.9 ± 2.1**	67.7 ± 1.9	66.5 ± 1	63 ± 2.2
ENRON-F	**57.8 ± 1.5**	57.1 ± 1.1	56.5 ± 1.8	54.3 ± 1.5
SLASHDOT-F	**44.4 ± 1**	43.3 ± 0.9	40.9 ± 2	32.5 ± 0.8
LLOG-F	**31.5 ± 1.4**	31 ± 1.5	30.5 ± 1.2	25.8 ± 1.9
WIPO	**77.5 ± 1.2**	**77.5 ± 1.2**	77 ± 0.8	71.3 ± 1.5
REUTERS	**76.8 ± 0.7**	76.7 ± 0.8	**76.8 ± 0.5**	69.8 ± 1.7
BIBTEX	35.7 ± 2.1	35.4 ± 1.9	**38.5 ± 1.1**	31.5 ± 1.5
BOOKMARKS	**19.8 ± 2.1**	19.2 ± 1.9	19.2 ± 2.3	16.6 ± 2.1
average	**57 ± 1.5**	56.7 ± 1.4	56 ± 1.4	50.1 ± 1.8

DATASET	BALANCED ACCURACY			
	ENSEMBLE	SINGLE	SVM	MLKNN
GENEBASE	**99.5 ± 0.3**	**99.5 ± 0.3**	99 ± 0.6	96 ± 0.9
CANCER	**74.1 ± 1.6**	74 ± 1.5	70 ± 1.6	63.1 ± 1.5
FINGERPRINT	**79 ± 1.2**	78.9 ± 1.1	77.6 ± 0.7	75.4 ± 1.2
ENRON-F	**74.7 ± 1**	74.2 ± 0.7	72.7 ± 1	71.9 ± 1.2
SLASHDOT-F	**72.4 ± 0.8**	71.9 ± 0.7	64 ± 0.8	60.3 ± 0.4
LLOG-F	**61.9 ± 0.7**	61.8 ± 0.7	59.5 ± 0.4	57.7 ± 0.7
WIPO	**84.5 ± 0.8**	84.4 ± 0.8	83.9 ± 0.5	79.4 ± 1.1
REUTERS	**84.2 ± 0.5**	**84.2 ± 0.6**	**84.2 ± 0.6**	78.7 ± 1.8
BIBTEX	**63.9 ± 0.5**	63.8 ± 0.4	63 ± 0.4	59.8 ± 0.5
BOOKMARKS	**57.4 ± 0.9**	57.2 ± 0.8	55.5 ± 0.7	54.6 ± 0.6
average	**75.2 ± 0.8**	75 ± 0.7	72.9 ± 0.7	69.7 ± 1

Figure 2: Winning model with respect to Hamming loss as the function of label balance and label correlation. Color scheme: red-ensemble MMCRF, orange-single MMCRF, green-ML-kNN, blue-SVM, gray-default classifier.



ACKNOWLEDGEMENTS

The work was financially supported by Helsinki Doctoral Programme in Computer Science (Hecse), Academy of Finland grant 118653 (ALGODAN), and in part by the IST Programme of the European Community under the PASCAL2 Network of Excellence, ICT-2007-216886.

Bregman divergence as general framework to estimate unnormalized statistical models

Michael U. Gutmann[1] Jun-ichiro Hirayama[2]

[1] Dept. of Computer Science and HIIT, Dept. of Mathematics and Statistics, University of Helsinki [2] Graduate School of Informatics, Kyoto University

www.cs.helsinki.fi/michael.gutmann

In short

We show that the Bregman divergence offers a rich framework to estimate statistical models which are possibly unnormalized, that is models where the normalization constant is not known in closed form:

- The framework works for continuous and discrete random variables.
- The framework includes several recent estimation methods such as noise-contrastive estimation, ratio matching or score matching.
- The framework provides links between unsupervised and supervised learning, as well as boosting.

Estimation of unnormalized models

- Goal: Given observed iid data $X = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, with $\mathbf{x}_i \sim p_d(.)$, find an estimate for p_d.
- Optimization approach: $\hat{p}_d = \mathrm{argmin}_{p_m \in \mathcal{P}_m} J(X, p_m)$ where J is a cost function and $\mathcal{P}_m$ the model family. We assume that $p_d \in \mathcal{P}_m$.
- Example: maximum likelihood for $p_d(\mathbf{u}) = p_m(\mathbf{u}; \boldsymbol{\theta}^\star)$, where $\boldsymbol{\theta}^\star$ denotes the "true" parameters,

$$\mathcal{P}_m = \{p_m^0(\mathbf{u};\boldsymbol{\theta})/Z(\boldsymbol{\theta}), p_m^0(\mathbf{u};\boldsymbol{\theta}) \geq 0, \tag{1}$$

$$Z(\boldsymbol{\theta}) = \int p_m^0(\mathbf{u};\boldsymbol{\theta})\mathrm{d}\mathbf{u}\} \tag{2}$$

$$J_{ll}(X, p_m) = -\sum_t \ln p_m^0(\mathbf{x}_t;\boldsymbol{\theta}) + \ln Z(\boldsymbol{\theta}) \tag{3}$$

What if you cannot compute the normalizing constant (partition function) $Z(\boldsymbol{\theta})$ exactly ?

– You cannot ignore the normalizing constant Z:
For $x_i \sim \mathcal{N}(0, \sigma^{\star 2})$ and $p_m^0(x;\sigma) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$ we had

$$\sum_{t=1}^T \nabla_\sigma \ln p_m^0(\mathbf{x}_t;\alpha) = \frac{1}{\sigma^3}\sum_{t=1}^T \mathbf{x}_t^2 \tag{4}$$

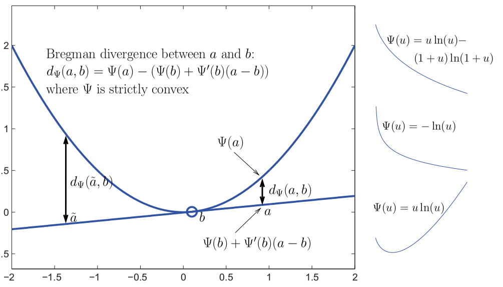
Stationary point is $\sigma \to \infty$.

– You cannot treat Z as parameter: $J_{ll}(\boldsymbol{\theta}) \to -\infty$ for $Z \to 0$.

– You could approximate the integral.

- The approach taken here: Avoid the complications that arise with Z. Choose $\mathcal{P}_m$ to consist of (a subset of) nonnegative functions. They should be summable but do not need to sum to one (unnormalized models).
- The presented work is about suitable cost functions for unnormalized models. The cost functions are based on the Bregman divergence.

Bregman divergence



- For two functions f and g: compute $d_\Psi(f(u), g(u))$ for all u in their domain, and take the weighted average.
- For f fixed, omit terms that depend only on f. This gives:

$$L(g) = \int \big[\underbrace{-\Psi(g) + \Psi(g)'g}_{S_0(g)} - \underbrace{\Psi(g)'}_{S_1(g)} f\big]\mathrm{d}\mu \tag{5}$$

All cost functions J considered in the paper derive from this equation.

Bregman div to estimate unnormalized models

- Choose f as a function "somehow related" to p_d and μ such that the integral in Eq. (5) can be computed as sample average, using data X.
- The choice discussed here on the poster (see paper for more!):

$$f = p_d/(\nu p_n), \qquad g = p_m/(\nu p_n), \qquad \mu\text{: cdf of } p_n \text{ multiplied by } \nu$$

p_n is a known (auxiliary) distribution and $\nu > 0$.

- This gives $L(g) = \nu\,\mathrm{E}\left[S_0\left(g(\mathbf{y})\right)\right] - \mathrm{E}\left[S_1\left(g(\mathbf{x})\right)\right]$ with $\mathbf{x} \sim p_d$, $\mathbf{y} \sim p_n$.
- Let p_n be a mixture of the original and a perturbation of the original data:

$$p_n(\mathbf{u}) = \alpha p_d(\mathbf{B}\mathbf{u} + \mathbf{v}) + (1-\alpha)p_d(\mathbf{u}) \tag{6}$$

where $\alpha > 0$ and $\mathbf{B}$ is an orthonormal matrix. $L(g)$ becomes $\tilde{L}(p_m; \mathbf{B}, \mathbf{v})$.

- *Ratio-matching*[1]: For binary data (±1), denote by $\mathbf{B}_i$ the matrix which flips bit i. For $S_0(u) = (1/2)u^2$, $S_1(u) = u$, $\alpha = \beta = 1/2$:

$$\tilde{L}(p_m; \mathbf{B}_i, 0) = 2\,\mathrm{E}\left(\frac{p_m(\mathbf{B}_i\mathbf{x})}{p_m(\mathbf{x}) + p_m(\mathbf{B}_i\mathbf{x})}\right)^2 - 1 \tag{7}$$

In ratio-matching, $\sum_i \tilde{L}(p_m; \mathbf{B}_i, 0)$ is minimized.

[1] A. Hyvärinen. Some extensions of score matching. Comp Statistics & Data Analysis, 2007

- *Score-matching*[2]: Assume $\mathbf{B} = \mathbf{I}$, $\mathbf{v}$: zero mean random variable with covariance matrix $\sigma^2\mathbf{I}$. If $\mathbf{x}$ is a continuous random variable,

$$\mathrm{E}_{\mathbf{v}}\,\tilde{L}(p_m; \mathbf{I}, \mathbf{v}) = \text{constant} + \sigma^2\alpha^2 S_1'(1)\,\mathrm{E}\left[\triangle_{\mathbf{x}} \ln p_m(\mathbf{x}) + \frac{1}{2}||\nabla_{\mathbf{x}} \ln p_m(\mathbf{x})||^2\right] + O(\sigma^3) \tag{8}$$

The term in the brackets is minimized in score matching.

Connection to supervised learning

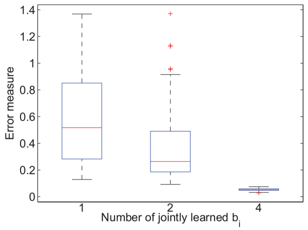
- Introduce a class variable C and consider $p_d(\mathbf{u}) = p(\mathbf{u}|C=1)$ and $p_n(\mathbf{u}) = p(\mathbf{u}|C=0)$.
- For $\nu = P(C=0)/P(C=1)$, the ratio $f = p_d/(\nu p_n)$ equals

$$f(\mathbf{u}) = \frac{p_d(\mathbf{u})}{\nu p_n(\mathbf{u})} = \frac{p(\mathbf{u}, C=1)}{p(\mathbf{u}, C=0)} = \frac{p(C=1|\mathbf{u})}{p(C=0|\mathbf{u})}. \tag{9}$$

- f can be used as discriminant function to classify between the two classes with minimal error rate: learning the ratio f means learning an optimal classifier.
- This generalizes noise-contrastive estimation[3] where logistic regression is used to estimate unnormalized models.

Connection to boosting

- Consider learning the log ratio $\ln f$: The objective L can be written as a function of $G(\mathbf{u}) = \ln p_m(\mathbf{u}) - \ln(\nu p_n(\mathbf{u}))$.
- For product-of-experts models, $\ln p_m$ factorizes: $G(\mathbf{u})$ is an additive model.
- Stepwise optimization of $G(\mathbf{u}) = \sum_i G_i(\mathbf{u})$ with $\Psi(u) = u\ln(u) - (1+u)\ln(1+u)$ corresponds to LogitBoost.
- Estimation in a stepwise manner is computationally lighter but leads to less accurate estimates:



Data distribution (ICA model):
$\ln p_d(\mathbf{x}) = \sum_{k=1}^4 -\sqrt{2}|\mathbf{b}_k^{\star T}\mathbf{x}| + (\ln|\det\mathbf{B}^\star| - 2\ln 2)$

Model:
$\ln p_m(\mathbf{x}; K, \boldsymbol{\theta}) = \sum_{k=1}^8 -\sqrt{2}|\mathbf{b}_k{}^T\mathbf{x}| + c$

Parameters $\boldsymbol{\theta}$ are $\mathbf{b}_k$ and c.

Error measure:
Compute 8×4 matrix $\mathbf{R} = \mathbf{B}^T(\mathbf{B}^\star)^{-1}$
Take Frobenius norm after subtraction of 4×4 identity matrix from upper block.

[2] A. Hyvärinen. Estimation of non-normalized statistical models using score matching. JMLR, 2005

[3] M.Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. AISTATS, 2010

Decoding Sensory Modalities from MEG Signals on the Basis of Spectrospatial Information

Jukka-Pekka Kauppi[1], Lauri Parkkonen[2], Riitta Hari[2], Aapo Hyvärinen[1]

[1]University of Helsinki, Helsinki, Finland [2]Aalto University, Espoo, Finland
e-mail: jukka-pekka.kauppi@helsinki.fi

Introduction

We use MEG-based decoding approach to investigate the organization of neural activity in the brain during natural hearing, vision, touch and rest. Unlike fMRI, MEG offers a possibility to investigate spectral signatures of neural activation relevant in information encoding.

We hypothesized that spectrospatial information present in MEG is useful in separating different stimulus categories from each other. To this aim, we designed four classifiers and evaluated their performance in a four-category decoding task.

Conlusions

❑Our results indicate that spectral information is useful in decoding stimulated sensory modalities from MEG data
❑Especially, relatively fine-grained spectrospatial information (utilized by our models C2 and C3) is useful
❑decoding based on unspecified spectral information (C1) did not result in results better than the baseline classifier not utilizing spectral information (C0)

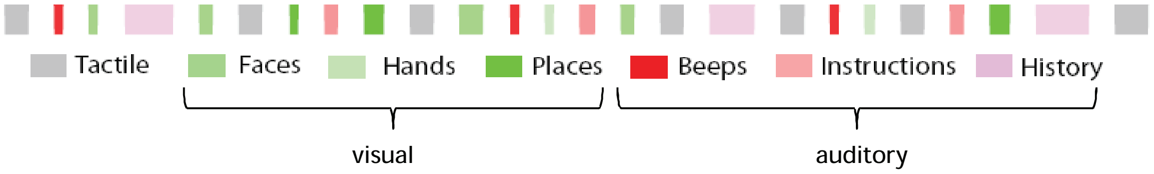


Figure 1. Stimulus sequence (12-min) [1]. The used categories were: 1. auditory, 2. visual, 3. tactile, 4. rest. White spaces denote rest blocks. Two sessions were recorded (training set and test set).

Experiment and preprocessing

❑Nine healthy adults exposed to 6-33-s blocks of auditory, visual and tactile stimuli that were interspersed with rest blocks (see Figure 1) [1]
❑Two 12-min sessions recorded: session 1 for the classifier training and session 2 for the performance evaluation

❑Short-time Fourier transform (STFT) applied to 2-sec MEG traces followed by independent component analysis [2]
❑C = 64 independent components (ICs) estimated
❑Frequency range from 5 to 30 Hz

Classifiers

❑Sparse multinomial logistic regression was used to perform classification of N spectral epochs
❑Four classifies (C1-C4) built based on varying degree of spectral information:
➢C0 did not use spectral information at all (features were the total energies of the ICs)
➢C1 used unspecific spectral information (features were the standard deviations of the spectra of the ICs)
➢C2 used category-wise spectrospatial information by treating each spectrospatial epoch as a matrix (C x N) [3]
➢C3 used spectrospatial information by estimating frequency coefficients for each IC with principal component analysis (PCA) prior to classification

Results

❑The (min/**mean**/max) classification accuracies of the 9 subjects for our models were:
➢C0: 0.24/**0.40**/0.63 (no spectral information)
➢C1: 0.25/**0.43**/0.68 (unspecific spectral information)
➢C2: 0.31/**0.50**/0.70 (category-wise spectral information)
➢C3: 0.35/**0.51**/0.64 (IC-wise spectral information)
❑the mean accuracies were clearly above the chance level (0.25) for each classifier
❑The classification rates of C2 and C3 were significantly higher than those from the baseline C0 (matched pair t-test; $p < 0.01$, uncorrected)
❑C0 > C1 not significant

References

[1] Ramkumar, P. (2011), 'Characterization of Neuromagnetic Brain Rhythms over Time Scales of Minutes Using Spatial Independent Component Analysis', Human Brain Mapping, in press, published online 13 Sep 2011.
[2] Hyvärinen, A. (2010), 'Independent Component Analysis of Short-time Fourier Transforms for Spontaneous EEG/MEG Analysis', Neuroimage vol. 49, no. 1, pp. 257-271.
[3] Dyrholm, M. (2007), 'Bilinear Discriminant Component Analysis', Journal of Machine Learning Research vol. 8, pp. 1097-1111.

Learning Topographic Representations for Linearly Correlated Components

Hiroaki Sasaki[1], Michael U. Gutmann[2], Hayaru Shouno[1] and Aapo Hyvärinen[2]

[1] Dept of Information and Communication Engineering, the University of Electro-communications

[2] Dept of Mathematics and Statistics, Dept of Computer Science and HIIT, University of Helsinki

Introduction

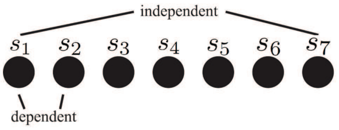
generative model

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{s}$$

$$\boldsymbol{x} = (x_1, x_2, \dots, x_d)^t \quad \boldsymbol{s} = (s_1, s_2, \dots, s_d)^t$$

- ICA is a statistical model to estimate independent non-Gaussian components.
- In ICA, the order of the sources cannot be estimated.

Relaxation of the assumption

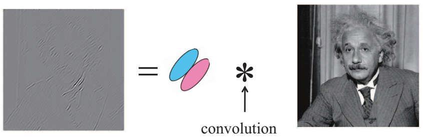


- In topographic ICA (TICA) proposed by Hyvärinen et al [1], the assumption in ICA was slightly relaxed, and the order can be estimated.
- We proposed a new model for topographic representations.
- Adjacent components in source signals are linearly correlated.

$$E\{s_i s_{i+1}\} > 0$$

- Distant components are as independent as possible.

Practical situation and motivation



- In practice, the outputs of two co-linear Gabor functions for natural image input can be linearly correlated.
- Topographic representations allow us to visualize the interrelation between components.
- Topography of natural stimuli may be related to cortical representations.

Model and its estimation

generative model

$$\boldsymbol{s} = \boldsymbol{u} \odot \boldsymbol{v}$$

The key properties of this generative model are:

- It generates super-Gaussian (sparse) components $\boldsymbol{s}$ [1].
- It generates correlated sparse components $\boldsymbol{s}$ when the components in $\boldsymbol{u}$ are independent but the adjacent components in $\boldsymbol{v}$ are linearly correlated.

Probability Density Function

$$p(\boldsymbol{s}) = \frac{1}{Z}\prod_{i=1}^{d} \exp(-|s_i|)\exp(-|s_i - s_{i+1}|)$$

Likelihood for the estimation of the model

$$J(\mathbf{W}) = J_1(\mathbf{W}) + J_2(\mathbf{W})$$

$$J_1(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d}|\boldsymbol{w}_i^t\boldsymbol{x}(t)| + \log|\det\mathbf{W}|$$

$$J_2(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d}|\boldsymbol{w}_i^t\boldsymbol{x}(t) - \boldsymbol{w}_{i+1}^t\boldsymbol{x}(t)|$$

$\boldsymbol{x}(t)$: t-th observation $\quad t = 1, 2, \dots, T$

$\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_d)^t = \mathbf{A}^{-1}$

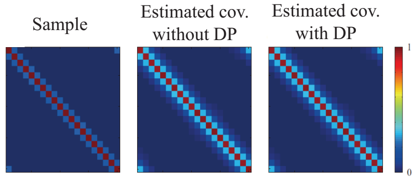
Flow of optimization

1. Estimation of $\mathbf{W}$ by the conjugate gradient method.
2. Optimization of order and signs.
3. Re-estimation of $\mathbf{W}$ by using optimized $\mathbf{W}_{ortho}$ as the initial input to the conjugate gradient method.

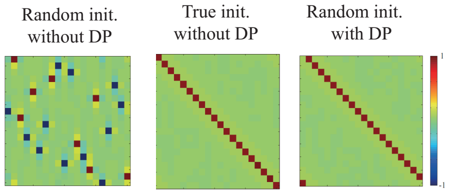
Validation on artificial data

- Artificial data are generated by two generative models.
- Preprocessing is to multiply a whitening matrix $\mathbf{V}$.
- Absolute values $|\cdot|$ are approximated as $\log\cosh(\cdot)$.

Covariance matrix

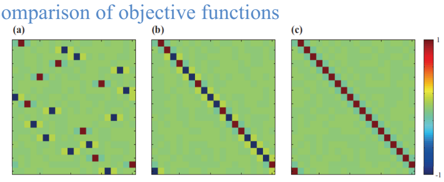


Matrix: $\mathbf{P} = (\mathbf{W}\mathbf{V})\mathbf{A}$



$J(\mathbf{W}) = -14.7014$ $J(\mathbf{W}) = -14.5973$ $J(\mathbf{W}) = -14.5973$

Comparison of objective functions



	(a)	(b)	(c)
$J(\mathbf{W}_{ortho})$	-15.9319	-15.9503	-15.8137
$J_1(\mathbf{W}_{ortho})$	-5.5700	-5.5700	-5.5700
$J_2(\mathbf{W}_{ortho})$	-10.3619	-10.3803	-10.2437

- Constraint: $\mathbf{W}_{ortho} = (\mathbf{W}\mathbf{W}^t)^{-0.5}\mathbf{W}$
- $J_1(\mathbf{W}_{ortho})$ is insensitive to the change of the order and signs.
- $J_2(\mathbf{W}_{ortho})$ shows the maximum value in the correct order and signs.

Formulation of a combinatorial optimization problem

$$\hat{\boldsymbol{k}}, \hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{k},\boldsymbol{c}} \underbrace{-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d} h(c_i\boldsymbol{w}_{k_i}^t\boldsymbol{x}_w(t), c_{i+1}\boldsymbol{w}_{k_{i+1}}^t\boldsymbol{x}_w(t))}_{J_2(\mathbf{W})}$$

$$h(a,b) = \log\cosh(a-b)$$

$$\boldsymbol{k} = (k_1, \dots, k_d) \quad k_i \in \{1, \dots, d\} \quad k_i \neq k_j \text{ for } j \neq i$$

$$\boldsymbol{c} = (c_1 \dots c_d) \quad c_i \in \{-1, 1\}$$

- A function $J_2(\mathbf{W})$ has a remarkable property: summation of functions of only two variables.
- The main problem can be divided into sub-problems.
- *Dynamic programming (DP)* would be an efficient method to solve the combinatorial optimization problem.

Algorithm inspired by dynamic programming

1. First, we fix $c_1 = 1$ and $k_1 = 1$. For $i = 2, \dots, d-1$, a function $f_{i+1}(c_{i+1}, k_{i+1})$ is defined as

$$f_{i+1}(c_{i+1}, k_{i+1}) = \max_{c_i, k_i}\left[f_i(c_i, k_i) - \frac{1}{T}\sum_{t=1}^{T} h(c_i\boldsymbol{w}_{k_i}^t\boldsymbol{x}(t), c_{i+1}\boldsymbol{w}_{k_{i+1}}^t\boldsymbol{x}(t))\right]$$

$$f_2(c_2, k_2) = -\frac{1}{T}\sum_{t=1}^{T} h(c_1\boldsymbol{w}_{k_1}^t\boldsymbol{x}(t), c_2\boldsymbol{w}_{k_2}^t\boldsymbol{x}(t))$$

Candidate functions for optimal values are also defined as

$$k_i^*(c_{i+1}, k_{i+1}), c_i^*(c_{i+1}, k_{i+1}) = \max_{c_i, k_i}\left[f_i(c_i, k_i) - \frac{1}{T}\sum_{t=1}^{T} h(c_i\boldsymbol{w}_{k_i}^t\boldsymbol{x}(t), c_{i+1}\boldsymbol{w}_{k_{i+1}}^t\boldsymbol{x}(t))\right]$$

2. For $i = d$, the optimal $\hat{c}_d$ and $\hat{k}_d$ can be obtained as

$$\hat{c}_d, \hat{k}_d = \arg\max_{c_d, k_d}\left[f_d(c_d, k_d) - \frac{1}{T}\sum_{t=1}^{T} h(c_d\boldsymbol{w}_d^t\boldsymbol{x}(t), c_1\boldsymbol{w}_1^t\boldsymbol{x}(t))\right]$$

3. From $i = d-1$ to $i = 2$, the optimal $\hat{\boldsymbol{k}}$ and $\hat{\boldsymbol{c}}$ can be found as

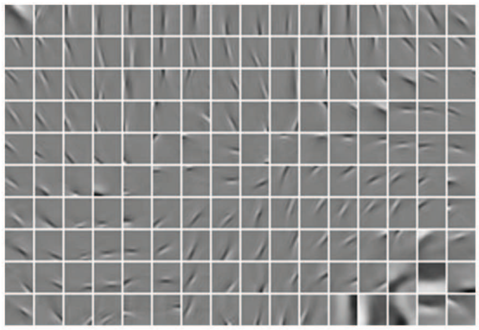
$$\hat{k}_i = k_i^*(\hat{c}_{i+1}, \hat{k}_{i+1}), \hat{c}_i = c_i^*(\hat{c}_{i+1}, \hat{k}_{i+1})$$

Experiments on real data

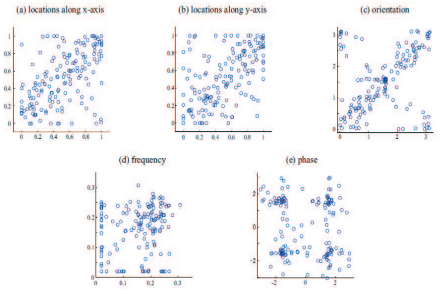
Natural image patches

- Data: 16 × 16 natural image patches.
- Preprocessing: the removal of DC components and whitening. The dimension is reduced to 160.

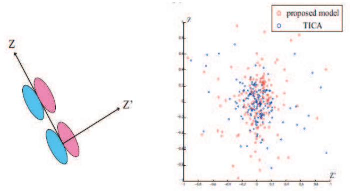
Estimated basis vectors



Dependency of adjacent basis vectors



Co-linearity of adjacent basis vectors



	proposed model	TICA
var(z)	0.1129	0.0593
var(z')	0.0655	0.0673

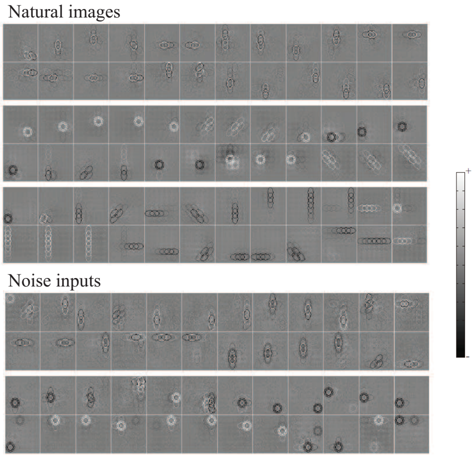
Complex cell outputs

- The outputs of complex cells are computed as

$$x'_k = \left(\sum_{x,y} W_k^o(x,y)I(x,y)\right)^2 + \left(\sum_{x,y} W_k^e(x,y)I(x,y)\right)^2$$

$$x_k = \log(x'_k + 1.0)$$

- For preprocessing, we remove the DC components and normalize the variances to one.



Connection to previous work

- Differences to TICA are:
  1. Adjacent components have linear correlation.
  2. Phases of basis vectors are non-random.
  3. Stronger co-linearity.
- For complex cells outputs, the results of previous work lacked topographic representations [2,3], while our model could estimate them reflecting the properties of natural images.

Conclusion

- We proposed a new statistical model to estimate topographic representations. In the model, adjacent components are linearly correlated, while distant components are as statistically independent as possible.
- To avoid local maxima in the likelihood, we proposed a new optimization method inspired by DP.
- The application to real data showed the emergence of new topographic representations.

Reference

[1] A. Hyvärinen et al, " Topographic independent component analysis ", Neural Computation, 2001.

[2] P.O. Hoyer et al, " A multi-layer sparse coding network learns contour coding from natural images", Vision Research, 2002.

[3] A. Hyvärinen et al, "Statistical model of natural stimuli predicts edge-like pooling of spatial frequency channels in V2 ", BMC Neuroscience, 2004

Statistical test for consistent estimation of causal effects in linear non-Gaussian models

Doris Entner[1], Patrik O. Hoyer[1], Peter Spirtes[2]

[1]Helsinki Institute for Information Technology & Department of Computer Science, University of Helsinki, Finland
[2] Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA

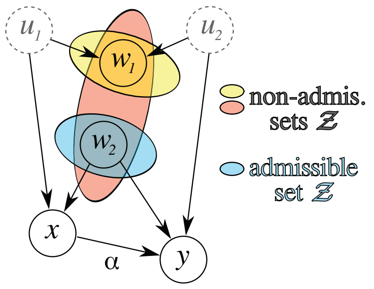
1. Motivation

Problem: Identify causal effects from *non-experimental* data
Challenge: Avoid inconsistent estimators due to confounding
Solution: 'Adjust for' a suitable set $\mathcal{Z}$ of observed variables, if such a set exists:

1. Underlying causal structure known:
   There exist graphical criteria and efficient algorithms to search for a suitable set $\mathcal{Z}$ (Pearl, 2009).
2. Underlying causal structure not known:
   - **Possible Solution 1 - Learn complete structure** (and use 1.)
     - In the Gaussian case only possible up to equivalence (Spirtes et al., 2000; Pearl, 2009)
     - In the linear non-Gaussian case possible if no latent variables (LiNGAM, Shimizu et al., 2006). If there are latent variables, only up to equivalence, computationally challenging (lvLiNGAM, Hoyer et al., 2008)
   - **Possible Solution 2 - Restrict to certain effects only**
     How to search for a suitable set $\mathcal{Z}$? → Subject of this poster (for linear models)

2. Basic Idea

Unknown generating structure:



Example in epidemiology:
x = risk factor
y = health indicator
$\mathcal{W}$ = set of general health conditions

GOAL: Obtain a consistent estimator of α, the causal effect of x on y, where x is the second last and y the last variable in the causal order.

IDEA: Search for a so called 'admissible' set $\mathcal{Z} \subseteq \mathcal{W}$ 'blocking' all information flow from x to y other than the direct one. 'Adjust for' this set to obtain a consistent estimator of α.

3. Background (Graphical Criterion, Admissible Set)

Back-door Criterion (Pearl, 2009)
A set $\mathcal{Z}$ fulfills the back-door criterion w.r.t. the ordered pair (x, y) if
- $\mathcal{Z}$ does not contain any descendants of x
- $\mathcal{Z}$ blocks (d-separates) every path between x and y that contains an arrow into x ("$x \leftarrow$")

A set $\mathcal{Z}$ fulfilling the back-door criterion is called *admissible*.

If $\mathcal{Z}$ is admissible then the causal effect α of x on y can be consistently estimated by *adjusting for* $\mathcal{Z}$ in the regression:

$$y = \hat{\alpha}x + \sum_{z \in \mathcal{Z}} c_z z + r_y$$

4. Model

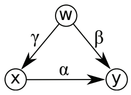
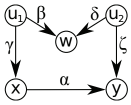
Assumptions:
- acyclic structure (unknown)
- linear relationships
- non-Gaussian, independent error terms e (unobserved)
- a set of unobserved variables $\mathcal{U}$
- a set of observed variables $\mathcal{W} \cup \{x, y\}$
- known partial causal order $\mathcal{W} \rightsquigarrow x \rightsquigarrow y$

Unknown equations:
(for generating structure in Box 2)

$$\begin{aligned} u_1 &= e_{u_1} \\ u_2 &= e_{u_2} \\ w_1 &= \beta u_1 + \gamma u_2 + e_{w_1} \\ w_2 &= e_{w_2} \\ x &= \zeta u_1 + \eta w_2 + e_x \\ y &= \alpha x + \kappa u_2 + \nu w_2 + e_y \end{aligned}$$

Observed variables:
w_1, w_2, x and y

5. Simple Example

Generating model w, x, y observed u_1, u_2 latent	$\mathcal{Z} = \emptyset$ $x = r_x$ $y = \hat{\alpha}x + r_y$	$\mathcal{Z} = \{w\}$ $x = bw + r_x$ $y = \hat{\alpha}x + cw + r_y$
	$\hat{\alpha}$ inconsistent $\mathcal{Z}$ not admissible $r_x \not\perp\!\!\!\perp r_y$	$\hat{\alpha}$ consistent $\mathcal{Z}$ admissible $r_x \perp\!\!\!\perp r_y$
	$\hat{\alpha}$ consistent $\mathcal{Z}$ admissible $r_x \perp\!\!\!\perp r_y$	$\hat{\alpha}$ inconsistent $\mathcal{Z}$ not admissible $r_x \not\perp\!\!\!\perp r_y$

6. Statistical Test for Consistency

Given a set $\mathcal{Z}$, estimate the two regressions using OLS

$$x = \sum_{z \in \mathcal{Z}} b_z z + r_x$$

$$y = \hat{\alpha}x + \sum_{z \in \mathcal{Z}} c_z z + r_y$$

If r_x is Gaussian → terminate without conclusion
If $r_x \perp\!\!\!\perp r_y$: $\hat{\alpha}$ is inferred to be a consistent estimator of α
If $r_x \not\perp\!\!\!\perp r_y$: $\hat{\alpha}$ is inferred to be an inconsistent estimator of α

Non-Gaussianity required since $\mathrm{cov}(r_x, r_y) = 0$, thus for Gaussian variables the residuals are always independent

7. Heuristics to Search for $\mathcal{Z}$

- Brute force - go through all possible sets $\mathcal{Z}$
- Forward selection - starting from the empty set, expand the "best" set from the previous round with the one variable which makes r_x and r_y most independent
- Backward elimination - starting from the full set, leave out the one variable of the "best" set from the previous round which makes r_x and r_y most independent

Main references

P.O. Hoyer, S. Shimizu, A.J. Kerminen, and M. Palviainen (2008). Estimation of causal effects using linear non-gaussian causal models with hidden variables. *IJAR 49: 362-378.*

J. Pearl (2009). *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2nd edition.

S. Shimizu, P.O. Hoyer, A. Hyvärinen, and A.J. Kerminen (2006). A linear non-gaussian acyclic model for causal discovery. *JMLR 7: 2003-2030.*

P. Spirtes, C. Glymour, and R. Scheines (2000). *Causation, Prediction, and Search.* MIT Press, 2nd edition.

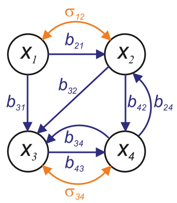
http://www.cs.helsinki.fi/u/entner/ConsistencyTest/ doris.entner@helsinki.fi

Causal Discovery for Linear Cyclic Models with Latent Variables

Antti Hyttinen ★ Frederick Eberhardt ★ Patrik O. Hoyer

HIIT ★ University of Helsinki ★ Washington University in St. Louis ★ MIT

Linear Cyclic Model with Latent Variables



$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ b_{21} & 0 & 0 & b_{24} \\ b_{31} & b_{32} & 0 & b_{34} \\ 0 & b_{42} & b_{43} & 0 \end{pmatrix}$$

$$\mathbf{\Sigma_e} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & 0 & 0 \\ \sigma_{12} & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & \sigma_{34} \\ 0 & 0 & \sigma_{34} & \sigma_4^2 \end{pmatrix}$$

$$\mathbf{x} := \mathbf{Bx} + \mathbf{e} \quad \Leftrightarrow \quad \begin{array}{lcl} x_1 & := & e_1 \\ x_2 & := & b_{21}x_1 + b_{24}x_4 + e_2 \\ x_3 & := & b_{31}x_1 + b_{32}x_2 + b_{34}x_4 + e_3 \\ x_4 & := & b_{42}x_2 + b_{43}x_3 + e_4 \end{array}$$

★ The model is parametrized by $\mathbf{B}$ and $\mathrm{cov}(\mathbf{e}) = \mathbf{\Sigma_e}$.

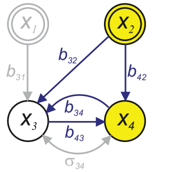
★ Behaviour at equilibrium:

$$\begin{array}{rcll} \mathbf{x}_t & := & \mathbf{B}\mathbf{x}_{t-1} + \mathbf{e} & \leftarrow \text{background conditions invariant} \\ \mathbf{x}_\infty & := & \mathbf{B}^\infty \mathbf{x}_0 + (\mathbf{I} + \mathbf{B} + \mathbf{B}^2 + \cdots)\mathbf{e} & \\ \mathbf{x}_\infty & := & (\mathbf{I} - \mathbf{B})^{-1}\mathbf{e} & \\ \mathbf{e} \sim \mathrm{N}(0, \mathbf{\Sigma_e}) & \Rightarrow & \mathbf{x}_\infty \sim \mathrm{N}(\mathbf{0}, (\mathbf{I}-\mathbf{B})^{-1}\mathbf{\Sigma_e}(\mathbf{I}-\mathbf{B})^{-T}) & \end{array}$$

A1 Self cycles are not identifiable from equilibrium data, so assuming $\forall i: b_{ii} = 0$.

A2 The model and all possible manipulated models are assumed stable: absolute of the eigenvalues of $\mathbf{B}$ and $\mathbf{B}^m$:s must all be less than 1.

Interventions & Experimental Effects



$$\mathbf{B}^m = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b_{31} & b_{32} & 0 & b_{34} \\ 0 & b_{42} & b_{43} & 0 \end{pmatrix}$$

$$\mathbf{\Sigma_e}^m = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & \sigma_{34} \\ 0 & 0 & \sigma_{34} & \sigma_4^2 \end{pmatrix}$$

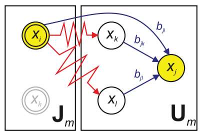
$$\mathbf{C}_x^k = (\mathbf{I} - \mathbf{B}^m)^{-1}\mathbf{\Sigma_e}^m(\mathbf{I} - \mathbf{B}^m)^{-T} = \overset{\text{independent randomizations}}{\downarrow}$$

$$\left(\begin{array}{cc|cc} 1 & \cdot & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ \hline t(x_1 \rightsquigarrow x_3 || \{x_1, x_2\}) & t(x_2 \rightsquigarrow x_3 || \{x_1, x_2\}) & v_3^2 & \cdot \\ t(x_1 \rightsquigarrow x_4 || \{x_1, x_2\}) & t(x_2 \rightsquigarrow x_4 || \{x_1, x_2\}) & v_{34} & v_4^2 \end{array}\right)$$

$$\underset{\text{experimental effects}}{\uparrow} \qquad \underset{\text{between observed}}{\uparrow}$$

$$\begin{array}{rcl} \mathbf{t}(\mathbf{x_2} \rightsquigarrow \mathbf{x_4} || \{\mathbf{x_1}, \mathbf{x_2}\}) & = & \text{Regression coefficient of } x_2 \text{ on } x_4 \\ & = & \text{Sum of all open paths from } x_2 \text{ to } x_4 \\ & = & b_{42} + b_{43}b_{32} + b_{43}b_{34}b_{42} + \cdots \\ & = & (b_{42} + b_{43}b_{32})(1 + b_{43}b_{34} + \cdots) \\ \text{nonlinear} \rightarrow & = & \dfrac{b_{42} + b_{43}b_{32}}{1 - b_{43}b_{34}} \\ & = & b_{42} + \dfrac{b_{32} + b_{34}b_{42}}{1 - b_{43}b_{34}} b_{43} \\ \text{linear} \rightarrow & = & \mathbf{b_{42}} + \mathbf{t}(\mathbf{x_2} \rightsquigarrow \mathbf{x_3} || \{\mathbf{x_1}, \mathbf{x_2}\})\mathbf{b_{43}} \end{array}$$

Linear Equations



$$t(x_i \rightsquigarrow x_j || \mathbf{J}_m) = b_{ji} + \sum_{x_k \in \mathbf{U}_m \setminus x_j} t(x_i \rightsquigarrow x_k || \mathbf{J}_m)\, b_{jk}$$

Method

1. Input covariance matrices $\mathbf{C}_x^1, \cdots \mathbf{C}_x^k$.
2. Form linear equations

$$\begin{pmatrix} \mathbf{K}_1 & & \\ & \mathbf{K}_2 & \\ & & \ddots \end{pmatrix} \begin{pmatrix} b_{12} \\ \vdots \\ b_{1n} \\ \hline b_{21} \\ \vdots \end{pmatrix} = \begin{pmatrix} t(x_2 \rightsquigarrow x_1 || \bullet) \\ \vdots \\ t(x_n \rightsquigarrow x_1 || \bullet) \\ \hline t(x_1 \rightsquigarrow x_2 || \bullet) \\ \vdots \end{pmatrix}$$

3. Solve for direct effects b_{ij}.
4. Get the covariances of the error terms σ_{ij} from an experiment where both x_i and x_j are observed with the formula

$$\sigma_{ij} = [(\mathbf{I} - \mathbf{B}^m)\mathbf{C}_x^k(\mathbf{I} - \mathbf{B}^m)^T][i, j]$$

Identifiability & Underdetermination

Identifiability theorem

Given a sequence of experiments the model $(\mathbf{B}, \mathbf{\Sigma_e})$ is fully identified by the method **if and only if** for each **ordered** pair of variables (x_i, x_j) there is

★ an experiment where x_i is **intervened on** and x_j is **observed** (Pair Condition), and

★ another experiment where both x_i and x_j are **observed** (Covariance Condition).

1. Say we have done experiments intervening on variables $\{x_1\}$, $\{x_1, x_2\}$, $\{x_3\}$.

$$\mathrm{PC}: \begin{pmatrix} \cdot & \times & \checkmark & \times \\ \checkmark & \cdot & \checkmark & \times \\ \checkmark & \checkmark & \cdot & \times \\ \checkmark & \checkmark & \checkmark & \cdot \end{pmatrix}, \quad \mathrm{COV}: \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \checkmark & \cdot & \cdot & \cdot \\ \checkmark & \checkmark & \cdot & \cdot \\ \checkmark & \checkmark & \checkmark & \cdot \end{pmatrix}$$

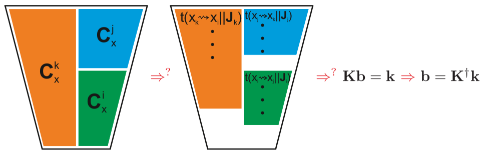
2. Which parameters are identified in the general case?

$$\mathbf{B}: \begin{pmatrix} \cdot & \times & \times & \times \\ \times & \cdot & \times & \times \\ \times & \times & \cdot & \times \\ \checkmark & \checkmark & \checkmark & \cdot \end{pmatrix}, \quad \mathbf{\Sigma_e}: \begin{pmatrix} \times & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot \\ \times & \times & \times & \cdot \\ \times & \times & \times & \checkmark \end{pmatrix}$$

★ Generally, for identifying b_{ji}, pair condition must be satisfied for all pairs $(\bullet, j)$.

★ If pair condition for (i, j) is not satisfied, then b_{ji} is never identified.

Completeness


$\mathbf{C}_x^k$
$\mathbf{C}_x^j$
$\mathbf{C}_x^i$
$\Rightarrow^?$
$t(x_k \rightsquigarrow x_l || \mathbf{J}_k)$
$t(x_i \rightsquigarrow x_l || \mathbf{J}_i)$
$t(x_j \rightsquigarrow x_l || \mathbf{J}_j)$
$\Rightarrow^? \mathbf{Kb} = \mathbf{k} \Rightarrow \mathbf{b} = \mathbf{K}^\dagger\mathbf{k}$

Completeness theorem

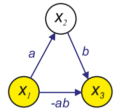
Given the data covariance matrices from a set of experiments, for determining the direct effects b_{ji}, the identifiability condition (Pair Condition) of the approach is **necessary** for any method.

This hinges on the fact that if two different direct effects matrices $\mathbf{B}$ and $\widehat{\mathbf{B}}$ produce the same experimental effects in a given set of experiments, the models $(\mathbf{B}, \mathbf{\Sigma_e})$ and $(\widehat{\mathbf{B}}, (\mathbf{I} - \widehat{\mathbf{B}})(\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Sigma_e}(\mathbf{I} - \mathbf{B})^{-T}(\mathbf{I} - \widehat{\mathbf{B}})^T)$ can be shown to produce the same covariance matrices for those experiments as well.

Assuming Faithfulness

Faithfulness in linear models

Any independence relation between variables is not the result of several exactly cancelling pathways.

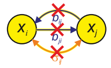


For every experimental dataset

1. Run a search for finding independencies. Add constraint equations from skeleton rule:

Skeleton rule

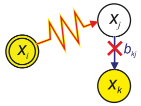
$$\frac{x_i \perp x_j \mid S \qquad x_j \notin \mathbf{J}_m}{b_{ji} = 0}$$



2. Add more constraint equations from orientation rules:

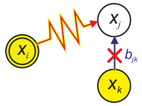
Orientation rule 1

$$\frac{t(x_i \rightsquigarrow x_k || \mathbf{J}_m) = 0 \qquad t(x_i \rightsquigarrow x_j || \mathbf{J}_m) \neq 0}{b_{kj} = 0}$$



Orientation rule 2

$$\frac{x_i \perp x_k | x_j \qquad t(x_i \rightsquigarrow x_j || \mathbf{J}_m) \neq 0}{b_{jk} = 0}$$



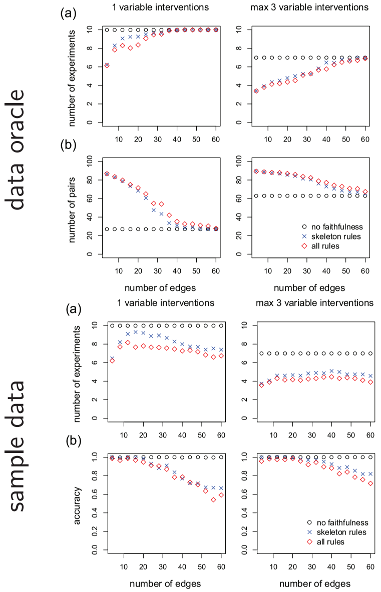
3. Take into account the additional structure found when selecting the next experiment.

Experiment Selection

1. Select the experiment that satisfies the pair condition for most new pairs.
2. If any parameters are identified, consider the pair condition for the corresponding pairs as satisfied.

Test Results

Selecting the experiment in such a way that the model is learned accurately with the fewest number of experiments.



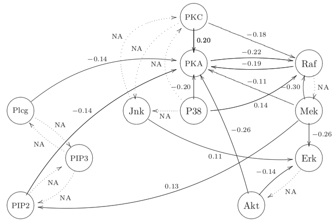
★ Fewer experiments are needed with sparse graphs.

★ More structure is discovered earlier on with sparse graphs.

★ With denser graphs the accuracy gets worse.

Sachs et al Flow Cytometry Data

Learning as much of the structure as possible given only 5 experiments, intervening on $\{\}$,$\{\mathrm{Mek}\}$,$\{\mathrm{PIP2}\}$,$\{\mathrm{Akt}\}$ and $\{\mathrm{PKC}\}$.



★ Pair condition was satisfied for only $40/110$ of the pairs, yet when assuming faithfulness most of parameters have been identified.

Summary

★ Method for learning linear cyclic models with latent variables using randomized experiments.

★ Complete with regard to search space and assumptions.

★ Necessary and sufficient identifiability condition.

★ Underdetermination characterized.

★ Faithfulness incorporated.

★ R-code available.

Noisy-OR Models with Latent Confounding

Antti Hyttinen, Frederick Eberhardt, Patrik O. Hoyer
University of Helsinki & HIIT & Washington University in St. Louis

Summary

We examine the identifiability of causal models with latent confounding, given a set of experiments in which subsets of the observed variables are subject to interventions.

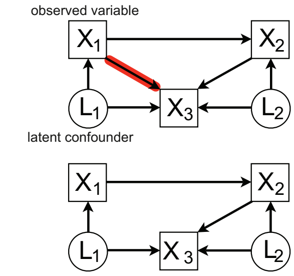
In general identifiability is impossible on the basis of experiments where only few variables are subject to intervention per experiment, which is often the case.

Identifiability is possible for a class of causal models whose conditional probability distributions are restricted to a 'noisy-OR' parameterization.

Identifiability is preserved under an extension of the noisy-OR CPD that allows for negative influences.

Several learning algorithms are introduced and tested for accuracy, scalability and robustness.

1. On the Identifiability of Causal Models with Latent Confounding



Passive observational data or experiments intervening on only a few variables at a time are generally insufficient to identify the **parameters** and the **structure** of a causal model with latent confounding.

For example, the two graphs on the left imply the exact same independences in single intervention experiments and when passively observed.

Furthermore, there exist parameterizations for the two graphs that produce the exact same distributions in those situations as well.

Thus, the presence of the **red** direct link cannot be determined unless both X_1 and X_2 are subject to an intervention in the same experiment.

2. Noisy-OR Model with Latent Confounding

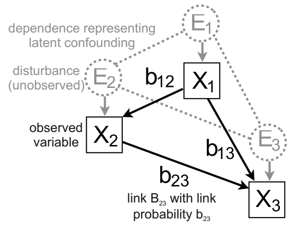
Structural Equation Model

$$X_1 := E_1$$
$$X_2 := (B_{12} \wedge X_1) \vee E_2$$
$$X_3 := (B_{13} \wedge X_1) \vee (B_{23} \wedge X_2) \vee E_3$$

Binary random variables X_1, X_2 and X_3 are observed. Links B_{12}, B_{23} and B_{13} and disturbances E_1, E_2 and E_3 are all unobserved binary random variables, introducing noise to the simple OR expressions.

Conditional Probability Distributions Links are independently distributed with model parameters $b_{12} = P(B_{12} = 1)$, b_{13} and b_{23}.

$$P(X_1 = 0|E_1) = (1 - E_1)$$
$$P(X_2 = 0|E_2, X_1) = (1 - E_2)(1 - b_{12})^{X_1}$$
$$P(X_3 = 0|E_3, X_1, X_2) = (1 - E_3)(1 - b_{13})^{X_1}(1 - b_{23})^{X_2}$$



Latent Confounding Latent confounding is represented by an arbitrary distribution $P(E_1^3)$ (total of 2^3 parameters). Any latent confounding (restricted by the noisy-OR CPD) can be presented through E_1, E_2 and E_3.

Joint Distribution

$$P(X_1^3) = \sum_{E_1^3} P(X_1|E_1)P(X_2|X_1, E_2)P(X_3|X_1, X_2, E_3)P(E_1^3)$$

Data Generation Draw a sample of disturbances E_1^3 from $P(E_1^3)$, links B_{12}, B_{13}, B_{23} from their independent distributions, and determine X_1,X_2 and X_3 from the SEM equations.

Context Specific Independence Property Noisy-OR CPDs have the following property.

$$(X_1 \perp\!\!\!\perp E_2 \,||\, X_1) \Rightarrow (X_1 \perp\!\!\!\perp E_2 \mid X_2 = 0 \,||\, X_1)$$

If parents X_1 and E_2 of variable X_2 are independent in some context (here when intervening on X_1), then additionally conditioning on their common child $X_2 = 0$ does not destroy this independence. This is evident from the SEM equations, if $X_2 = 0$, then $E_2 = 0$ and $(B_{12} \wedge X_1) = 0$, thus the value of E_2 does not provide any additional information about the value of X_1.

3. Identifiability

The parameters of any three variable model can be identified from single intervention experiments and passive observational data.

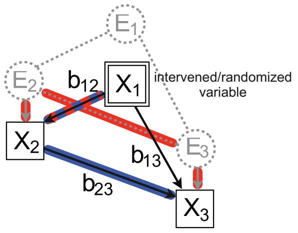
Step 1 Find a causal order from the ancestral relationships directly observed in the experiments and rename variables such that the causal order is X_1, X_2, X_3.

Step 2 Estimate link probability b_{12} by Cheng's causal power formula, using the intervention on X_1 to make E_2 independent of X_1.

$$b_{12} = \frac{P(X_2 = 1||X_1 = 1) - \overbrace{P(X_2 = 1||X_1 = 0)}^{\substack{X_2 = 1 \text{ caused by} \\ \text{its other causes}}}}{\underbrace{1 - P(X_2 = 1||X_1 = 0)}_{\text{renormalization}}}$$

Similarly, estimate b_{23} by intervening on X_2.

$$b_{23} = \frac{P(X_3 = 1||X_2 = 1) - P(X_3 = 1||X_2 = 0)}{1 - P(X_3 = 1||X_2 = 0)}$$



Step 3 Estimate the link probability b_{13} by additionally conditioning on $X_2 = 0$ s.t. the **blue** indirect path is intercepted.

$$b_{13} = \frac{P(X_3 = 1|X_2 = 0||X_1 = 1) - P(X_3 = 1|X_2 = 0||X_1 = 0)}{1 - P(X_3 = 1|X_2 = 0||X_1 = 0)}$$

The context specific independence property guarantees that the **red** path remains intercepted.

Step 4 Estimate the noise distribution from the passive observational data by solving a matrix equation:

$$\overbrace{\begin{bmatrix} \ddots & \vdots & \vdots & \vdots \\ \cdots & (1 - b_{12}) & 0 & 0 \\ \cdots & 0 & (1 - b_{13})(1 - b_{23}) & 0 \\ \cdots & b_{12} & b_{13} + b_{23} - b_{13}b_{23} & 1 \end{bmatrix}}^{P(X_1^3|E_1^3)} \overbrace{\begin{bmatrix} \vdots \\ P(E_1^3 = 101) \\ P(E_1^3 = 110) \\ P(E_1^3 = 111) \end{bmatrix}}^{P(E_1^3)} = \overbrace{\begin{bmatrix} \vdots \\ P(X_1^3 = 101) \\ P(X_1^3 = 110) \\ P(X_1^3 = 111) \end{bmatrix}}^{P(X_1^3)}$$

The matrix on the left is lower triangular with a nonzero diagonal, and thus invertible.

All parameters of a noisy-OR model with latent confounding are identified from the combination of a **passive observational data set** and a set of experiments where **for each ordered variable pair (X_i, X_j) there is an experiment where X_i is randomized and X_j is observed**. This condition is also necessary.

4. Learning Algorithms

Efficient Conditioning Conditioning reduces the effective sample size for estimating the link probabilities. However, if it happens in step 2 (above) that $b_{12} = 0$ or $b_{23} = 0$, then the **blue** path does not exist and conditioning on X_2 is unnecessary when estimating b_{13}. The correct conditioning sets for each link can always be determined based on links already estimated. In addition, the experimental data can also be taken into account when estimating $P(E_1^3)$.

EM-algorithm For up to eight variables, the model can also be learned using a version of the EM-algorithm.

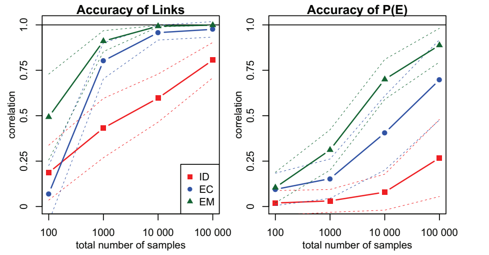
5. Extension to Negative Influences

In noisy-OR models, the parents X_1 and X_2 being ON has a positive effect on their child X_3 being ON. However, the noisy-OR parameterization can be extended to also allow for negative influences:

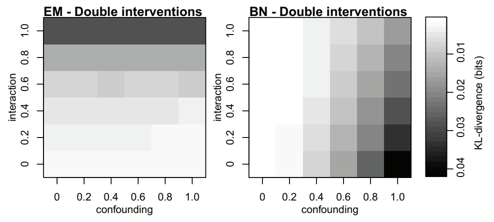
$$X_3 := E_3 \vee (B_{13} \wedge \widetilde{X}_1) \vee (B_{23} \wedge \widetilde{X}_2),$$

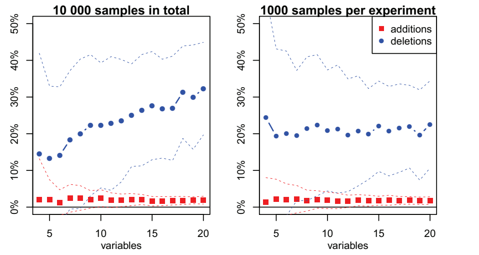
where for positive/generative causes $\widetilde{X}_i = X_i$ and for negative causes $\widetilde{X}_i = \neg X_i$. Now $X_1 = 0$ can cause $X_3 = 1$. The context specific independence property and the identifiability of the model are preserved.

6. Simulations



Accuracy Accuracy of the learning algorithms with increasing sample sizes. EM is most accurate, EC beats the algorithm based on the identifiability proof (ID).





Scalability Structural errors when using the EC-algorithm on models with different sizes. Some statistically insignificant links are deleted.

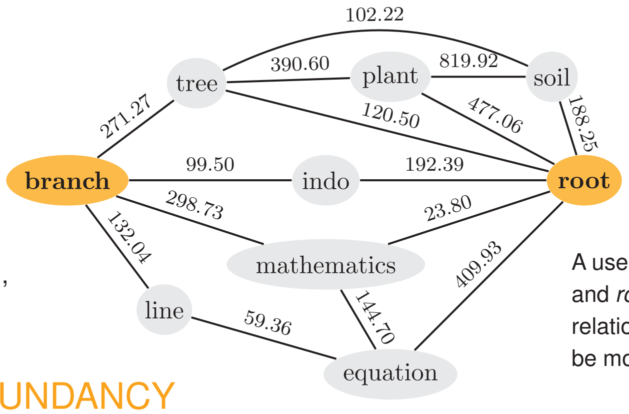
Robustness Models were learned from single intervention and passive observational data, generated by a 'noisy-interactive-OR' model while the amount of latent confounding and interaction of the parents was varied. The shade of each square represents the average predictive accuracy in double intervention experiments. Lighter shades indicate better results. Standard Bayesian Network without hidden variables (BN) predicts accurately when there is little confounding, noisy-OR (EM) predicts accurately when there is only little interaction.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
TIETOJENKÄSITTELYTIETEEN LAITOS
INSTITUTIONEN FÖR DATAVETENSKAP
DEPARTMENT OF COMPUTER SCIENCE

RELEVANT AND NON-REDUNDANT OBJECT RETRIEVAL

Laura Langohr and Hannu Toivonen

We address a setting of information retrieval where the user specifies query objects and the problem is to identify other objects that are relevant with respect to the query objects, but non-redundant with respect to each other.
Consider, as example the graph on the right, where nodes represent terms (objects), edges relations between them, and weights word co-occurrences within sentences.



A user who wants to know how *branch* and *root* are related might know some relations. Other relations again might be more interesting.

RELEVANCE AND NON-REDUNDANCY

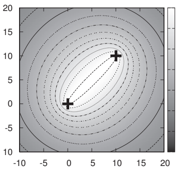
RELEVANCE

The *relevance* of an object $u \in V$ with respect to a positive query object $q \in V$ is defined as their proximity:

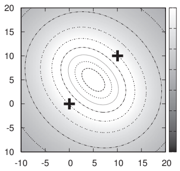
$$rel_P(u,q) = s(u,q) = 1/d(u,q).$$

The relevance of object u with respect to a set $Q_P \subset V$ of query objects is defined as the inverse of the p-norm with $\alpha \geq 1$:

$$rel_P(u,Q_P) = \Big(\sum_{q \in Q_P} d(u,q)^{\alpha} \Big)^{-\frac{1}{\alpha}}.$$



$\alpha = 1$



$\alpha = 4$

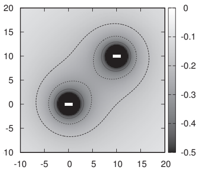
IRRELEVANCE

The irrelevance (or negative relevance) of an object u with respect to a negative query object $\bar{q} \in V$ is defined as their proximity:

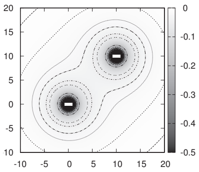
$$rel_N(u,\bar{q}) = s(u,\bar{q}) = 1/d(u,\bar{q}).$$

The irrelevance of object u with respect to a set $Q_N \subset V$ of negative query objects is defined as the sum of similarities raised to the power of $\beta > 1$:

$$rel_N(u,Q_N) = \sum_{\bar{q} \in Q_N} d(u,\bar{q})^{-\beta} = \sum_{\bar{q} \in Q_N} s(u,\bar{q})^{\beta}.$$



$\beta = 1$



$\beta = 2$

NON-REDUNDANCY

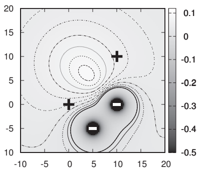
Given a set $R \subset V$ of (retrieved) objects, the *redundancy* of R is defined by

$$red(R) = \sum_{\substack{u,v \in R \\ u \neq v}} d(u,v)^{-\beta} = \sum_{\substack{u,v \in R \\ u \neq v}} s(u,v)^{\beta}.$$

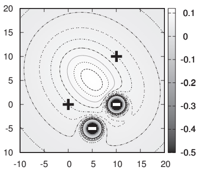
RELEVANCE AND NON-REDUNDANCY

We define the overall relevance and non-redundancy of a set of objects $R \subseteq V$ as

$$REL(R,Q_P,Q_N) = \sum_{u \in R} rel_P(u,Q_P) - rel_N(u,Q_N) - red(R).$$



$\alpha = 4, \beta = 1$



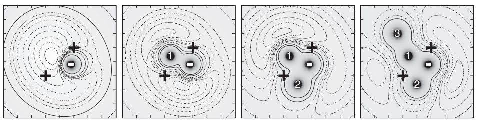
$\alpha = 4, \beta = 2$

ALGORITHMS AND EXPERIMENTS

FINDING RELEVANT AND NON-REDUNDANT OBJECTS

Greedy algorithm

```
1. Repeat until a sufficient number of representatives has been retrieved:
   1.1. Find the most relevant object $r$ w.r.t. $Q_P$ and $Q_N$
   1.2. Output $r$ and add it to $Q_N$
```



Iterative algorithm

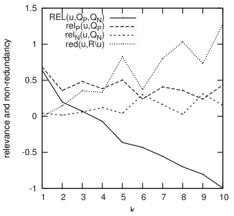
```
1. Get an initial solution $R$ of $k$ objects (e.g. random)
2. Repeat while $R$ changes:
   2.1. Find the optimal swap of any object $r$ in $R$ to any object not in $R$
   2.2. If the swap improves the result, implement it
```

EXPERIMENTS

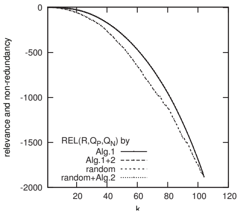
Word relations and senses: The proximity is measured by word co-occurence within sentences.

Q_P	branch & root	bank	star
1.	tree	reserve	planet
2.	indo	river	trek
3.	mathematics	gaza	cluster
4.	line	credit	sirius
5.	equation	international	movie

Biomedical graph and probabilistic node proximities.



Overall relevance (solid line) and its factors (relevance, dashed; irrelevance, short dashed; non-redundancy, dotted line).



Overall relevance of set R_k of top k nodes obtained by both algorithms and random ranking. (The lines corresponding to the algorithms are indistinguishable.)

CONCLUSION

- Both algorithms produce a good set of objects, with high relevance and low redundancy.
- The greedy algorithm seems to also work well for any top k objects.
- The iterative algorithm could in our experiments produce only marginally better results than the greedy ranking.
- What application can you think of?

University of Helsinki, Faculty of Science
Department of Computer Science, Discovery Group

Corpus-Based Generation of Content and Form in Poetry

Jukka M. Toivanen, Hannu Toivonen, Alessandro Valitutti, and Oskar Gross

Background

We present a method for generation of novel poetry. The main idea is to use two different corpora, on one hand, to provide semantic content for new poems, and on the other hand, to generate a specific grammatical and poetic structure. The approach uses text mining methods, morphological analysis, and morphological synthesis to produce poetry in Finnish.

Computational poetry is a recent and challenging research area of computer science, at the cross section of computational linguistics and artificial intelligence. Poetry is one of the most expressive ways to use verbal language. For this reason, computational generation of texts recognizable as good poems is difficult to achieve. Nevertheless, several interesting research systems have been developed for the task (see e.g. Manurung, Ritchie, and Thompson 2000; Gervás 2001; Manurung 2003). These systems vary a lot in their approaches, and many different computational and statistical methods are often combined in order to handle the linguistic complexity and creativity aspects.

Methods

- The topics and semantic coherence of generated poetry are controlled by using a simple word association network which is automatically constructed from the background corpus using word co-occurence analysis.
- The grammar, including the syntax and morphology of the generated poetry, is obtained in an instance-based manner from a given grammar corpus. Instead of explicitly representing a generative grammar of the output language, we copy a concrete instance from an existing sentence or poem but replace the contents.
-The current poetry generation procedure can now be outlined as follows:
- **(1)** A topic is given (or randomly chosen) for the new poem. The topic is specified by a single word.
- **(2)** Other words associated with the topic are extracted from the background graph.
- **(3)** A piece of text of the desired length is selected randomly from another corpus.
- **(4)** Words in the text are analyzed morphologically (part of speech, singular/plural, case, verb tense, clitics etc.).
- **(5)** Words in the text are substituted independently, one by one, by words associated with the topic. The substitutes are transformed to similar morphological forms with the original words. The replacement is applied only when a word in the expansion of the topic can be transformed to the correct morphological form. In other cases the original word is left intact.
- **(6)** After all words have been considered, the novelty of the poem is measured by the percentage of replaced words. If the poem is sufficiently novel it is output.

Examples

Some example poems generated with the system are presented below with their rough English translations:

Kuinka hän leikki silloin
uskaliaassa, uskaliaassa kuiskeessa
vaaleiden puiden alla.
Hän oli kuullut huvikseen
kuinka hänen kuiskeensa
kanteli helkkeinä tuuloseen.

How she played then
in a daring, daring whispering
under the pale trees.
She had heard for fun
how her whispering drifted as
jingle to the wind.

Laula se ukkoseksi suurten sielujen
luo onnien, jumalin
hoiloa sielu!
Kirkkaus!
Tuo korkea herra huus:
"Pyhä hehkuko meiltä soinut ois?"

Sing it to the thunder of great souls
to the blisses, with gods
sing out, you soul!
Brightness!
That high lord yelled:
"I wonder if the holy blaze would have
chimed from *us.*"

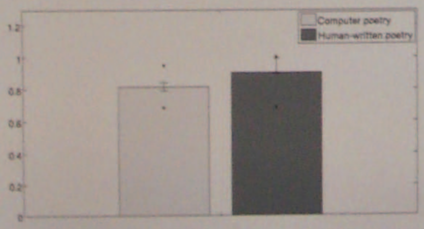
Vaaleassa kourassa
sopusuhtaisessa kourassa ovat nuput niin kalpeita
kuvassasi lepää lapsikulta jumala.

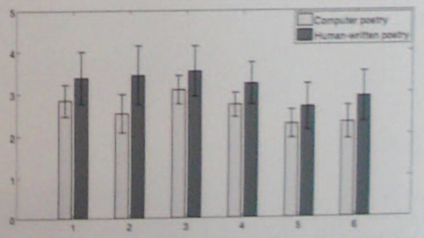
In a pale fist
in a well-balanced fist, the buds are so pale
in your image lies a dear child god.

Evaluation

We evaluated poetry using a panel of twenty randomly selected subjects. Each subject independently evaluated a set of 22 poems: One half were human-written poems from the grammar corpus and the other half computer-generated. The subjects were not explicitly informed that some of the poems were computer-generated. The first question to answer was if the subject considered the piece of text to be a poem or not, with a binary yes/no answer. The figure shows also standard deviations of the answers and averages for the best and worst poems in the both groups.

Then each text was evaluated qualitatively along six dimensions: (1) How typical is the text as a poem? (2) How understandable is it? (3) How good is the language? (4) Does the text evoke mental images? (5) Does the text evoke emotions? (6) How much does the subject like the text? These dimensions were evaluated on the scale from one (very poor) to five (very good). The whiskers indicate the standard deviations of the answers.





Discussion

It may be questioned whether the current approach exhibits creative behaviour, and whether the system is able to produce poetry that is interesting and novel with respect to the text that is used as the basis of new poetry. The empirical results indicate that this is the case. First, the generated poems are usually very different from the original texts. Second, the preliminary evaluation results show that some of the generated texts were rated to be quite untypical, even though recognized as poems. The pleasantness and language quality of these poems were still judged to be relatively high. According to these observations we think that at least some of the system's output can be considered to be creative. Thus, the system could be argued to automatically piggyback on linguistic conventions and previously written poetry to produce novel and reasonably high quality poems.

References

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. Journal of Knowledge-Based Systems 14(3-4):181-188.
Manurung, H. M., Ritchie, G., and Thompson, H. 2000. Towards a computational model of poetry generation. In Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science, 79-88.
Manurung, H. 2003. An evolutionary algorithm approach to poetry generation. Ph.D. Dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Correspondence to:
jukka.toivanen@helsinki.fi,
hannu.toivonen@cs.helsinki.fi

Ambiguous Lexical Resources For Computational Humor Generation

Alessandro Valitutti

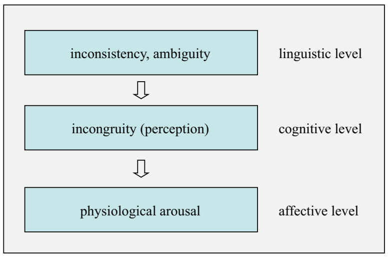
University of Helsinki – Department of Computer Science and HIIT

alessandro.valitutti@cs.helsinki.fi

Introduction

- This work is aimed to investigate to what extent it is possible to perform a feasible use of ambiguous lexicon in computational humor.
- The first core of a lexical database, characterized as an extension of WORDNET 3.1 (Fellbaum, 1998), was developed in order to collect ambiguous terms in the English lexicon for be employed as resource for humor generation.

Humor and Ambiguity



inconsistency, ambiguity — linguistic level

⇩

incongruity (perception) — cognitive level

⇩

physiological arousal — affective level

Humor is a way to induce mirth, a specific emotion. In verbal humor, linguistic ambiguity can affect the cognitive state, through the violation of expectation. The corresponding state of can increase the level of emotional arousal and contribute to the humorous effect.

Humor and Lexical Ambiguity

Double-Edged Words (DEW)

A DEW can be characterized by the following attributes:

1. **WORD** is the lexical unit (e.g. a single word or a phrase).
2. **AMBIGUITY** is a list of two or more "meanings" associated to the WORD.
3. **DEPTH** expresses the different typicality of the two meanings. For example, a two fold ambiguity will be associated to a main meaning (called *surface meaning,* with depth 1) and a secondary meaning (called *hidden meaning,* with depth 2).
4. **SLANT** is a set of additional semantic labels associated to the hidden meaning, and characterizing it as potentially humorous. Slant labels can be used to emphasize the humorous role of hidden meaning. For example, slant labels can be selected in order to evoke ridiculous trait of people.

Homonymic DEWs

- Homonymy is defined as the relation between words that share the same spelling and pronunciation but have different meanings.
- In WordNet each word meaning is represented by a set of synonyms (synset) and associated to a specific ID in the database. Each word is associated to one of more *senses* (i.e. ranked synsets).
- Homonymic DEWs are words in WordNet with at least two senses.
- The sense number expresses the DEPTH attribute. A list of 24167 DEWs was extracted from WordNet 3.1.

Homophonic DEWs

- Homophony is defined here as the relation between words that are phonetically identical (complete *homophones)* or similar (partial *homophones)* but with different spelling.
- The algorithm for the measure of the phonetic distance is a specific implementation of the Levenshtein distance.
- A measure of the above described phonetic distance was calculated for all pairs of words in WordNet, in order to collect sets of homophones. A number of 5400 total homophonic sets and 23050 partial homophonic sets were filtered.

Idiomatic DEWs

- Idiomatic ambiguity is a specific type of ambiguity between literal and figurative language. Idioms are defined here as multiword expressions whose meaning cannot be inferred by the meaning of the component words. The idiomatic meaning of a word is the meaning associated to the idiom in which the word is included.
- A manual annotation of WordNet was performed in order to identify lexical idioms (i.e. idioms consisting of a composed word).
- The collection includes 3541 WordNet synsets.
- For each idiomatically ambiguous word, the surface meaning (or *literally meaning)* was defined as its first sense in Word- Net, and the hidden (or *idiomatic meaning)* as the first sense in the idiom in which the word is included.

Double-Edged WordNet (DEWN)

Items are defined according to three different possible types of lexical ambiguity:

1. **Homonymy** is defined as the relation between words that share the same spelling and pronunciation but have different meanings (e.g. tablet)
2. **Homophony** is defined here as the relation between words that are phonetically identical (complete homophones) or similar (partial homophones) but with different spelling (e.g. show/shop).
3. **Idiomatic ambiguity** is a specific type of ambiguity between literal and figurative language. Idioms are defined here as multiword expressions whose meaning cannot be inferred by the meaning of the component words. The idiomatic meaning of a word is the meaning associated to the idiom in which the word is included. (e.g. cat/rain)

Examples

Punning Riddles

How do you define a pig?
It is a stout-bodied short-legged omnivorous policeman.

In order to obtain this joke, the homonymic DEW "pig" was selected. The definition (in the form of answer) is the gloss of the default meaning (i.e. first WordNet sense of the corresponding noun), in which the word "animal" was substituted by the first synonym ("policeman") of the hidden meaning (i.e. third WordNet sense).
The creation of a punning riddle starting from a "lexical core" is inspired to the JAPE system (Binsted and Ritchie, 1994), in which the joke is generally based on a couple of phonetically similar words.

An analogue example is:

Who is a working girl?
A young streetwalker who is employed.

Funny Acronyms

CPU = Celibate Professing Untied (from "Central Processing Unit")

This type of acronym generation is modeled on the HAHAcronym system (Stock and Strapparava, 2002). The acronym is generated through the replacement of each word in the original expansion (Cen*tral Processing Unit)* according to phonetic similarity ("processing" vs. "professing") and semantic opposition ("computer" vs. "religion").

The following "hand-made" example, instead, cannot be generated with the present resource because it involve a model of the ambiguity propagated at the phrase level:

IBM = Interpreting Bible Machines
(from the original *International Business Machines)*

Variation of Familiar Expressions

A chapel a day keeps the malefactor away.

This example is based on the FEVER program (Valitutti, 2011). The pun is obtained through two word replace- ments in which both phonetic similarity and domain slanting (RELIGION) constraints were applied.

Instead the following hand-made expression cannot be generated without a model describing the ambiguity at the sentence level:

An onion a day keeps everyone away.

Conclusion

- Exploration of the connection between computational humor and automatic discovery
- Distinction between heuristic creativity and narrative creativity
- Distinction between ambiguity and "slanting"
- Definition and collection of ambiguous lexical units – DEWN
- Integration of existing humor generators

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

The use of weighted graphs to investigate the large-scale evolution of metabolism

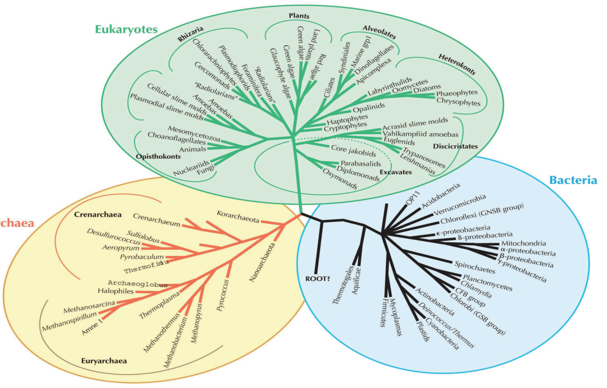
(ongoing work)

Fang Zhou [a] joint work with Ross D. King [b] and Hannu Toivonen [a]
[a] Department of Computer Science and HIIT, University of Helsinki, Finland
fang.zhou @ cs.helsinki.fi, hannu.toivonen@cs.helsinki.fi
[b] Manchester Interdisciplinary Biocentre, The University of Manchester, UK
ross.king @ manchester.ac.uk

Abstract

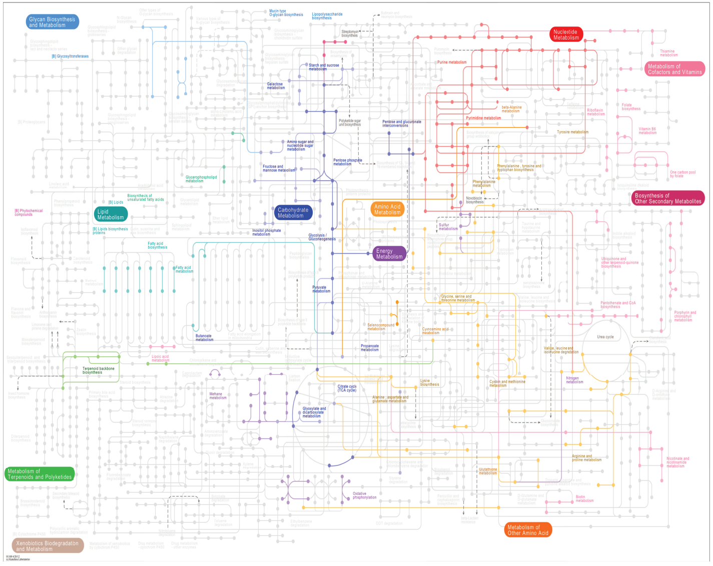
We are interested in better understanding the evolution of metabolic biodiversity in bacteria- *Archaea* and the *Eubacteria*. To investigate this question, we introduce the use of weighted graphs to integrate large amounts of genomic data. We propose three ways of measuring the importance of enzymes, and apply the weighted graph compression method to measure the correlation between two kingdoms.

Tree of life



Metabolism

Each species has a different metabolism. Given a large number of species, how can we compare their metabolisms?

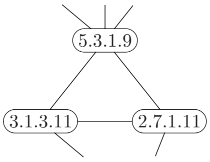

Glycan Biosynthesis and Metabolism
Nucleotide Metabolism
Metabolism of Cofactors and Vitamins
Lipid Metabolism
Carbohydrate Metabolism
Amino Acid Metabolism
Energy Metabolism
Biosynthesis of Other Secondary Metabolites
Metabolism of Terpenoids and Polyketides
Xenobiotics Biodegradation and Metabolism
Metabolism of Other Amino Acid

Weighted metabolic network

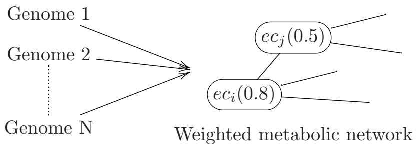
Goal: Integrate different species metabolisms into one graph.

Solution:

Step 1: Represent the meta-metabolic network as a graph with enzymes as nodes. Two enzymes are connected with an edge if they catalyze reactions that share metabolites.



Step 2: Assign weights to enzymes based on how frequent they are in the species.

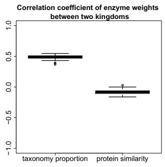


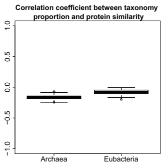
Enzymes with high weights are ubiquitous, and those with low weights rarely occur.

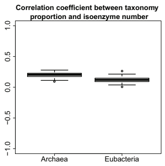
Such a weighted graph summarizes the information in the set of these instantiations of the meta-metabolic network.

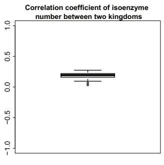
Three ways of weighting

- Taxonomy proportion = $\dfrac{\text{no. of genomes that contain the enzyme}}{\text{no. of genomes}}$.
- Protein similarity = average similarity of protein sequences of the enzyme.
- Average isoenzyme number = average number of isoenzymes.









We found that the important enzymes, determined by their existence frequency, in *Archaea* are also important in *Eubacteria*. However, this importance is not presented by neither sequence conservation nor average number of isoenzyme.

Correlation between two kingdoms

We apply the weighted graph compression method to compress the metabolic network utilizing enzyme weights, and decompress the compressed graphs to enable direct comparison between them.

	Compression ratio				
	0.01	0.03	0.05	0.07	0.09
Distance	53.94	59.54	62.22	64.14	65.47

Mean distance between compressed graphs of *Archaea* and *Eubacteria* at different compression ratios.

Results show: more compression actually gives a smaller distance.

Future work

(1) Apply the graph compression method to compare the importance of pathways in the different kingdoms.

(2) Extract an approximate ancestor metabolism, which is a connected subgraph with enzymes that are common to both kingdoms.

(3) Use simulations to produce a null-model for pathway evolution.